Essays on Matchings and Contests

by

Rohan Chowdhury

Submitted in fulfilment of the requirements for the

Degree of

Doctor of Philosophy in Economics

to

The Adam Smith Business School

College of Social Sciences

University of Glasgow

January, 2022

This page is intentionally left blank.

To Ma

This page is intentionally left blank.

Abstract

This thesis is composed of two parts. Part I contributes to the matching literature and Part II contributes to the contest literature.

Part I: A set of indivisible objects must be allocated among a group of agents, or agents from one side of the market must be matched to agents on the other. Each agent has their own preference over these objects, or over the agents on the other side. These preferences exhibit an underlying structure, motivated by real-world examples such as the refugee settlement problem, the job rotation problem and others. We study design of procedures to match agents to objects/agents, without resorting to randomization devices or monetary transfers. Three concerns play a central role in this design: incentive compatibility, efficiency and fairness. For each setting we consider, we appropriately formalise what we mean by these terms.

Part II: A competitive league coach (team manager) must manage the energy pool of players in the team over the course of the entire season. High energy usage increases the chance of winning the current game, but also increases the risk of developing accumulated fatigue or injuries in later games. We explore this "winning in the short term" vs "saving energy for the future" dilemma that the coach faces over a long season.

Contents

Abstract **iv**

List of Figures **viii**

List of Tables **x**

Acknowledgements **xi**

0 Introduction **1**

I Matching Theory **12**

1 Allocating agents on a line with birth dates **13**

1.1 Introduction 13

1.1.1 Informal discussion of design goals 15

1.1.2 Three Assignment Rules 17

1.1.3 Brief summary of results 18

1.1.4 Related literature 19

1.1.5 Outline of the chapter 21

1.2 Model 21

1.2.1 Desirable properties of allocations 22

1.3 Assignment Rules 26

1.3.1 Natural assignment rules in the context 26

1.3.2 Desirable properties of Assignment Rules 28

1.4 Results 32

1.5 Concluding remarks 41

1.6 Appendix A - Proofs 43

2 Stable matchings with indifferences and the priority rule 50

2.1 Introduction 50

2.1.1 Relation to the existing literature 54

2.1.2 Outline of the chapter 57

2.2 Model 57

2.2.1 The Preference Structure 57

2.2.2 Notations and definitions 58

2.3 Stable and efficient matchings 61

2.4 Strongly stable matchings 70

2.5 Discussion and concluding remarks 74

2.6 Appendix B - Proofs 76

3 Dynamic job rotation: an impossibility result 81

3.1 Introduction 81

3.2 Model 83

3.2.1 Preferences over job allocation plans 84

3.2.2 Matchings 85

3.2.3 Job rotation priority structure 85

3.2.4 The criteria for matchings 86

3.2.5 Mechanisms 88

3.3 The impossibility result 89

3.4 Conclusion and further direction 92

II Contest Theory 95

4 Effort dynamics in a competitive league 96

4.1 Introduction 96

4.1.1 Related literature 100

4.1.2 Outline of the chapter 101

4.2 Base model 102

4.3 Analysis and Results 108
4.3.1 Expected Win Rate 110
4.3.2 Optimal Strategy 111
4.3.3 End Season 112
4.4 Extensions 113
4.4.1 Extension I : Field plays strategically 113
4.4.2 Extension II : Strategic play by the team in all quarters 119
4.4.3 Extension III: Strong and weak opponents 130
4.4.4 Comparison with Base Model 134
4.5 Concluding remarks 137
4.6 Appendix C 138
4.6.1 Proofs 138
4.6.2 Code used for Python simulations 141

Bibliography **149**

List of Figures

4.1 Expected Win Rate at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 10 , 40 , 50 , 60 , 70 110

4.2 Optimal Strategy at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 10 , 40 , 50 , 60 , 70 111

4.3 Expected Win Rate in the last five games (t). Colour codes: $t =$ 75 , 76 , 77 , 78 , 79 113

4.4 Optimal Strategy at the end of the season. Colour codes: $t =$ 75 , 76 , 77 , 78 , 79 113

4.5 Expected Win Rate at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 10 , 40 , 50 , 60 , 70 114

4.6 Optimal Strategy at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 10 , 40 , 50 , 60 , 70 115

4.7 Expected Win Rate at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 75 , 76 , 77 , 78 , 79 116

4.8 Optimal Strategy at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 75 , 76 , 77 , 78 , 79 117

4.9 Optimal strategies in three iterations. Going from left to right, optimal strategies in *Iteration 2* appears in the left most column, followed by *Iteration 3*, and finally *Iteration 4*. 118

4.10 Expected Win Rate and Optimal Action in Q1-Q3 at different stages of the season. Different shades denote different game (t), with colour codes: $t =$ 10 , 20 , 40 , 60 , 70. 122

4.11 Expected Win Rate at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 10 , 40 , 50 , 60 , 70 124

4.12 Optimal Energy usage in Q4 at different stages of the season. Colour codes: $t =$ 10 , 40 , 50 , 60 , 70 125

4.13 Expected Win Rate and Optimal Action in Q1-Q3 towards the end of the season. Different shades denote different game (t), with colour codes: $t =$ 75 , 76 , 77 , 78 , 79. 126

4.14 Expected Win Rate in the last five games (t). Colour codes: $t =$ 75 , 76 , 77 , 78 , 79 127

4.15 Optimal Strategy at the end of the season. Colour codes: $t =$ 75 , 76 , 77 , 78 , 79 127

4.16 Optimal energy usage in the first three quarters combined when the team starts strong. 128
4.17 Optimal Action in $Q4$ when the team starts strong ($e^P = 0.8$). 128
4.18 Optimal energy usage in the first three quarters combined when the team starts with average energy. 129
4.19 Optimal Action in $Q4$ when the team starts average ($e^P = 0.5$). 129
4.20 Optimal energy usage in the first three quarters combined when the team starts with average energy. 130
4.21 Optimal Action in $Q4$ when the team starts weak ($e^P = 0.2$). 130
4.22 Expected Win Rate at different stages of the season. Different shades denote different game (t), with the following colour codes: t = 5 , 15 , 40 , 60 , 75. 132
4.23 Optimal strategy at different stages of the season. Colour codes: t = 5 , 15 , 40 , 60 , 75. 133
4.24 Expected Win Rate. Colour codes: t = 75 , 76 , 77 , 78 , 79 134
4.25 Optimal Strategy at the end of the season. Different shades denote different game (t), with the following colour codes: t = 75 , 76 , 77 , 78 , 79 135

List of Tables

3.1 Preference profiles and priority structure for the proof of Theorem 3.3.1. 89

4.1 Summary of notations. 104

4.2 Number of Strong Opponents remaining at different stages of the season. 131
4.3 Number of Strong Opponents to face in remaining games 133
4.4 Comparison of *EWR* across the different models. 135
4.5 Comparison of optimal strategies across the different models. 136

Acknowledgements

This thesis has been possible due to the enormous amount of support and encouragement I have received over the last several years, and I would like to take this opportunity to express my gratitude to all those, who have in their own way, contributed to my ability to submit today.

First and foremost, I express my sincere gratitude to Prof. Arunava Sen and Prof. Debasis Mishra for introducing me to the wonderful world of game theory. Thank you for inspiring me and believing in me. I would not have embarked on this journey if it were not for you both.

It goes without saying that I could not have written this dissertation without the constant support and encouragement of my advisors Prof. Anna Bogomolnaia and Dr. John Levy. I owe my deepest gratitude to both of you. Thank you for being patient with me in answering my countless questions, suggesting valuable references, and always helping me see the light whenever I felt stuck. I would like to express my sincere gratitude to Prof. Hervé Moulin for his advice and guidance, which significantly helped me navigate the initial phase of my journey when I had little clue about how to conduct research. My sincere thanks go out to all three of you, for enabling me to realize the importance of asking the right questions, and for helping me recognize the ingredients of a good research project. These have played a big part in developing and shaping my taste and approach to research in general.

I would like to thank Prof. Takashi Hayashi and Prof. Indrajit Ray for agreeing to be my thesis examiners. I appreciate the opportunity to discuss my work with both of you, and to receive your insightful comments and suggestions.

I am thankful to the University of Glasgow for providing an ideal atmosphere and all institutional support over the course of my studies. I am appreciative of the scholarship and travel fund received from the College of Social Sciences for my PhD studies, and I feel very fortunate for the same.

I would like to thank Foivos, Damiano, and Ashraful for their comments and thoughtful suggestions that have greatly helped improve the quality and presentation of this dissertation.

I was fortunate to have great seniors like Li, Mo, Foivos, Seemanti, Hualin and Arsenii. Thank you for your advice and guidance on approaching the PhD journey. Thank you for brainstorming with me on several occasions: I have learnt a lot from those sessions.

I would like to express my deepest thanks to Sophie Watson for her immense help and support with all my queries over these years.

I am indebted to my wonderful friends in Glasgow with whom I have shared bread and beer and words and laughs. Thank you Dev, Damiano, Adhiraj, Ashraful, Foivos, Prateek, Swagatam, Laura, Helena, Natalia, Hualin, Elizaveta, Mo, Max, Seemanti, Nikolas, Jinglin, Ahn, Arthur, Arsenii, Vladimir, Jerome, Li, Johanna. As I sit down to recollect, many fond memories come to mind: the reading groups, the cafe trips, the gigs, the pool sessions to unwind after work, the dinners, the discussions and debates spanning a wide range of topics that certainly changed the world, the birthday celebrations at the office(s), the trips to the highlands, the indoor football sessions, and many more.

Music has always been very important to me in my life and has certainly played a big role in keeping my sanity in check during hard times. Thank you, Damiano and Foivos, for introducing me to new avenues that I will forever cherish.

I feel grateful for my friendship with Prachi, Swarnendu, Dhritiman, Pranabesh, and Abhinaba. Thank you all for always being a willing ear, for sharing your honest opinion, and for cheering me up through my tough times.

Thank you Helena for your love and invaluable companionship. Your positive energy and constant encouragement certainly played a tremendous role in my completing this arduous journey. I also thank you for proofreading parts of this dissertation.

Thank you Ma, for your unconditional support in all endeavors that I have undertaken throughout my life. It is not possible to list all the sacrifices that you have made in bringing me up as the person I am today. I have missed you, and I apologize for being so far, and for so long. This thesis is as much yours as it is mine (the errors are only mine).

Glasgow, January 2022

Affidavit

I declare that, except where explicit reference is made to the contribution of others, this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution.

Printed Name: *Rohan Chowdhury*

Signature:

This page is intentionally left blank.

Chapter 0

Introduction

The thesis is composed of two parts: Part I consists of three chapters on assignment and matching problems that are studied from the lens of mechanism design; and Part II consists of a single chapter that studies a dynamic contest model. The chapters are written as research papers, and are fully self-contained, with an individual literature review that discusses the relation to existing work.

Part I

Mechanism design has been an active and fruitful sub-field of economics since the 1970s. Since the seminal contribution of Hurwicz (1973) who introduced the key notion of incentive compatibility[1], this literature has shed light on and offered solutions to a wide range of real-world problems such as the design of radio spectrum auctions, the design of college admissions, kidney exchange programs, the national resident matching program, and many more.

Combining the points of view of social choice theory and game theory, the theory of mechanism design aims to design games (or, *mechanisms*) such that whenever rational and strategic players interact within the rules of the game, their interaction yields an outcome that is desirable to the mechanism designer. The mechanism designer (or, central planner) sets out with an objective in mind: for instance,

[1] A mechanism is incentive compatible if participants can achieve the best outcome for themselves just by acting according to their true preferences.

the objective can be utilitarian (maximization of the total utility of agents), Rawlsian (maximization of utility of the least well-off agent), etc. The designed mechanism induces a *game-form* whose equilibrium outcome is precisely the objective that the designer has set.

The primitives in a mechanism design problem are the set of possible outcomes or alternatives, and the preferences of agents over this set. These preferences are not known to the designer. Mechanism design problems can broadly be classified based on whether or not monetary transfers are allowed. Transfers serve as a means to redistribute utility among agents.

1. *Mechanism design without money*. In many important environments, money cannot be used to redistribute utility among the agents. This constraint typically arises from ethical and/or institutional considerations. Examples here are: political elections (Gibbard, 1973; Satterthwaite, 1975), kidney exchange programs (Roth et al., 2004), the stable matching problem (Gale and Shapley, 1962).

2. *Mechanism design with money*. A very well studied domain is the quasi-linear environment: here the utility function of agents takes a quasi-linear form (apart from a possibly non-linear component, there is a separable and linear dependence on money). Classical examples include: the auction of an item (Vickrey, 1961; Myerson, 1981), building a public project (Clarke, 1971; Green and Laffont, 1977), the bilateral trade problem (Myerson and Satterthwaite, 1983).

While there are many papers that study very general preference domains, an extensive body of literature has investigated various restrictions of such domains that are motivated by real-world observations. Many general results typically do not remain applicable in restricted domains. For instance, in the context of mechanism design without money, two seminal papers by Gibbard (1973) and Satterthwaite (1975) introduce a very general model (with at least three alternatives) and prove that it is impossible to find a non-dictatorial mechanism that is incentive-compatible. But this impossibility vanishes when the preference domain is restricted to simpler contexts such as (i) model with one-dimensional policy space over which agents have single-peaked preferences (Moulin, 1980), where the policy space can represent, e.g., the political spectrum in the case of elections, or a line in the case of facility location problems; (ii) allocation and exchange of indivisible goods such as transplant organs, public housing, etc. for which agents have single-unit demand (Roth, 1982; Svensson, 1999; Pápai, 2000).

When one shrinks a preference domain, the class of incentive-compatible mechanisms expands. Therefore, even if the larger domain admits "nice" mechanisms, it becomes important to devise ways to select between already existing mechanisms and new mechanisms that become available in the smaller domain. As an example, consider the problem of allocating indivisible goods to agents so that each agent receives at most one good. In this model, Ma (1994) proved that the top trading cycle mechanism (see, Shapley and Scarf, 1974 for a description) is the only mechanism that is strategy-proof, efficient, and individually rational when the domain consists of all linear preferences. However, when the domain is restricted to single-peaked preferences, this result is no longer true: in fact, Bade (2019) discusses another mechanism that satisfies an even stronger incentive property called obvious strategy-proofness (for a definition, see Li, 2017) without sacrificing the other two. Furthermore, in smaller domains existing/known mechanisms can satisfy additional desirable properties, and it becomes important to point those out.

In the spirit of the above discussion, Part I of this thesis investigates particular domain restrictions in the context of different matching models. The study contributes to the literature on mechanism design without money. Specifically, we will be interested in problems where (i) a set of heterogeneous indivisible objects must be allocated among a group of agents (one-sided matching); or (ii) agents from one side of the market must be matched to agents on the other (two-sided matching). While these models have been previously studied allowing for very general preferences, the focus here will be to restrict attention to smaller preference domains that are motivated by important and frequently arising real-world applications. We elaborate on the specifics of each restricted domain below in the context of the chapters. Three concerns play a central role in designing the mechanisms: incentive compatibility[2], efficiency, and fairness. We discuss important trade-offs that come to light. While some of our results are reminiscent of previous observations in the literature, we also present results where we show that some traditional facts do not hold anymore, instead, we observe new phenomena that arise. We discuss them in substantial detail in the individual chapters.

The three chapters in this part focus on purely deterministic settings. The first two chapters study models that are static, while the third chapter studies a dynamic matching framework. The chapters share the following features:

* There are two sides of the market: a set of agents on one side; and either a set of indivisible objects, or another set of agents on the other.

* Every agent is endowed with a preference list over the set of available objects/agents on the other side.

* Preferences of agents on one side of the market are restricted to have a well-behaved structure. This is guided by motivating examples which are discussed below in the context of every chapter.

* The objects are sometimes endowed with a priority order over agents, that must be respected in the following sense: if an agent i has priority over another agent j for some object o, then agent j should not receive object o if agent i receives an object inferior to object o.

There are several matchings possible for any given instance of the problem. Is there one among them that is better than others in some objective sense? If there is one, how do we find it? Is it better than others in one criterion but not as good

[2]Typically in the form of stability or strategy-proofness.

in another but equally important one? Is there one that is better than others in a number of ways?

To answer questions of this kind, we study the problem of a central agency (or, designer) who is tasked with designing a mechanism that will systematically match agents to objects/agents; by meeting a set of desirable goals (defined formally as mathematical axioms). We think of this central agency to not have any selfish interests, but instead, to work solely to improve social welfare in some way.

A brief description of the chapters in Part I as well as the examples that motivate them are outlined next.

Chapter 1

Consider a scenario where there are several large-scale governmental projects that must be undertaken, and there are firms/organizations that would like to take up such projects. Oftentimes, these projects can be ranked in terms of desirability. Nevertheless, firm-specific capacity constraints (stemming from technological limitations, deadlines requirements, etc.) might prevent the firm from competing for the best-ranked projects. Next, consider another scenario where users want a slot at a bottleneck facility: researchers waiting to use a supercomputer, ships waiting to load/unload cargo at a port, airlines waiting to use the runway for take-off/landing, electric vehicles waiting to be re-charged at a power station, and so on. Users arrive at different times but wish to be served as early as possible once they arrive.

In these examples, one side of the market consists of agents (firms/users) while the other side consists of objects (projects/slots), and agents have preferences that are structured in the following sense: all agents agree with the exogenous ranking of objects, but all objects in the rank order above a certain threshold (specific to an agent) are not feasible to the corresponding agent. Chapter 1 lays down a framework incorporating these ingredients.

The primary welfare determining factor in this setting is the *delay* faced by an agent, which captures the amount of disutility the agent suffers by not receiving his/her best object. Accordingly, an important goal for the designer might be to minimize the aggregate delay faced by all agents. Minimizing aggregate delay might come at the expense of some agents. Therefore, in light of fairness, one might instead wish to minimize the maximum delay faced by any agent; or better yet, minimize lexicographically the vector of delays. In conjunction with these goals, we study other design objectives that have to do with ensuring Pareto efficiency and achieving various incentive properties that deter participating agents from being dishonest and trying to game the system to their advantage, either on their own accord or in collaboration with other agents. In particular, we explore the compatibility of different combinations of these goals. We also propose different mechanisms and evaluate each of them with respect to these goals.

Chapter 2

This chapter extends the model in Chapter 1 in two ways — first, by endowing each project with a preference list; and second, by allowing several projects to have the same rank. Since we endow projects with preferences, we now refer to projects also as agents instead of objects. Projects are once again exogenously ranked, but this time multiple projects can have the same rank. Firms have preferences that have a nested structure just as in Chapter 1, but this time around, their preferences can naturally have indifferences. This is because multiple projects can now have the same rank. On the other hand, preferences of projects are allowed to be very general. To summarize, in Chapter 2 we study what is called a two-sided one-to-one matching model with indifferences.

The leading example here is the refugee matching problem: the projects' side of the market is comprised of refugee families of various sizes, and the firms' side consists of benevolent hosts who are willing to shelter refugees in their homes in exchange for a monetary reward (which is increasing in family size) from the government. Thus, in essence, there is an exogenous ranking over refugee families based on family size and hosts prefer larger families, but can only accommodate families of size up to the number of beds they have to offer. Refugee families can have arbitrary preferences (owing to economic prospects, cultural or religious concerns, location factors, etc.) over hosts.

The designer's problem is to choose a matching that would guarantee that agents would accept the proposed matching, and would not try to individually renegotiate it. We follow the traditional approach in the literature and insist on *stability* as the important selection criterion. A stable matching sustains itself: no pair of agents in a stable matching have incentives to break away from it. The notion of a blocking pair makes precise the conditions for agents to have the aforementioned "incentives". Since the preference domain allow indifferences, based on three natural definitions of a blocking pair (see, Irving (1994)), there are three natural ways to define stability: *weak stability*, *strong stability*, and *super stability*. We study the first two notions only since they are widely regarded as the most appropriate criterion for a practical matching scheme when there are indifferences in the preference lists. Apart from stability, we also impose that the proposed matching be Pareto efficient. Any attempt to improve the welfare of an agent in an efficient matching always comes at the expense of another.

Chapter 2 explores the implication of these properties on the structure of the resulting matching. In particular, we show that stable and efficient matchings in this framework are hierarchical by nature: they can be thought of as an outcome of projects being arranged in a queue order, with each project receiving one of their best firm in turn, from the set of remaining firms.

Chapter 3

The motivation of this chapter comes from the organization of *job rotation*: the well-known practice of moving employees through a range of different tasks/jobs, in order to promote their interest, experience, and motivation. The use of such a strategy is widespread across both private and public sectors (Osterman, 1994; Ostrom, 1990; Berkes, 1992). Contrary to the static models studied in previous chapters, Chapter 3 adopts a dynamic matching framework to study the rotation problem and contributes to the literature on — dynamic priority-based allocation of indivisible objects.

The different tasks (or, job functions) have priorities over employees, which can be thought to have been derived from observable employee characteristics such as their skill-set, work experience, etc. The employer would like to respect these priorities, i.e., he/she would like to give the employee who has higher priority over another for a given job, greater access to that job. But, the employer would also like to encourage job rotation. To achieve this dual goal, the employer can use the temporal dimension: as long as both employees are inexperienced in some given job, the employee with the higher priority gets more access to the job; but if the agent with the higher priority already has previous experience in the job, the other agent gets a chance instead to garner some experience in the job despite having a lower priority for it to begin with. This requirement is encoded as a job-rotation-priority structure that becomes a design goal to satisfy.

Since the model is dynamic, we define appropriate dynamic counterparts of very standard and well-studied goals/axioms concerning efficiency, stability, and incentive properties of the mechanism. We prove a negative result here: it is impossible to satisfy the job-rotation requirement while being as efficient as possible, and also at the same time ensuring that the mechanism is immune to strategic manipulations by the employees.

Part II

This part of the thesis contributes to the literature on Contest Theory. Contests and tournaments are ubiquitous in social and economic landscapes. In a traditional contest model, economic agents expend costly nonrefundable resources to vie for a limited number of prizes, and they are rewarded for "getting ahead" of their opponents instead of their absolute performance metrics. Such competitive interactions are common in a diverse range of contexts — promotion competitions between firms to increase market share, R&D races, sports, rent-seeking activities, political campaigns, college admissions, and so on.

Since the seminal contribution of Tullock (1967), a large literature has analyzed strategic aspects of contest-like activities. The basic contest model has been enriched by introducing different modelling variations such as information asymmetry between competitors, dynamic structure, multi-battle confrontations, etc. An important modelling dimension concerns the choice of winner-selection rule: the rule that takes as input an effort (resource) profile[3] and selects one or more winners. Such a rule can be deterministic, as in all-pay auction models (Baye et al., 1996); or stochastic, e.g., (i) ratio-form contest success functions (Tullock, 1967), (ii) rank-order tournaments with noise (Lazear and Rosen, 1981). The rich literature (with many variations like the ones stated above) makes use of these basic rules in some form (tailored to each specific setting) in order to ascertain the contest winner(s). The main focus in the literature has therefore been to compare final performances (effort choices) of the participants in a contest. In contrast, our model (described below) makes an attempt to study the effort dynamics during an ongoing contest, with the focus being to look at the impact of (observable) progress at an interim point in the contest on the effort choice thereafter. The motivation comes from team sports such as basketball, netball, rugby, or handball leagues; where the *score-difference* (which is observable) between competing teams can change by large margins very rapidly and frequently. A summary of the model is briefly outlined next.

Chapter 4

In this chapter, we are interested in the decision problem of a competitive league coach, who wishes to maximize the success of the team over the course of the entire season. The resource at the coach's disposal that the coach has to carefully expend is the pool of players in the team; more specifically, the cumulative energy levels

[3] A profile of efforts is a list consisting of the effort choices of all participants in the contest.

of these players. During any game, players are exposed to frequent high-intensity movements like sprinting, jumping, acceleration, deceleration, abrupt change of direction, etc. This can lead to acute and accumulated fatigue, and therefore calls for effective management of the team energy level. Resting or limiting minutes of players, especially the “star” players in the team, is indeed something coaches frequently consider. On the other hand, if the players do not exert sufficient effort in any given game, it may jeopardize the team’s success in the short run.

The rest of star players resulting from coaches’ optimal strategies is also of considerable interest to team owners, investors in the league, etc. Fans are more willing to pay for quality competition that features the star players, and hence revenue maximization considerations require designing the season’s structure so that coaches optimizing will lead to more opportunities for the fans to see the best players. Hence it is important for those with a vested interest in the revenue from the sport’s fans to know how the incentives set for the teams will affect the time the best players spend on the courts and pitches.

The model in Chapter 4 explores the trade-off between the two above-mentioned competing forces in a long season: “winning in the short term” vs “saving energy for the future”. Specifically, the model looks at energy dynamics “late” in a game and the impact of score-difference between the competing teams up to that point in the game on the energy-savings choice of the coach thereafter. The evolution of the aggregate energy level of the team and the winner of any given game are both determined stochastically. We model coaching decisions as a Markov Decision Process and find that, indeed, saving energy is optimal for the coach throughout most of the season. This conclusion is robust to different model extensions, in particular, against a field of teams whose coaches also employ similar strategies.

This page is intentionally left blank.

Part I

Matching Theory

Chapter 1

Allocating agents on a line with birth dates

Abstract

In this chapter, we consider the problem of assigning agents to slots arranged on a line. Agents have birth dates on the line and want a slot as soon as possible once they are born. We carry out an axiomatic study of this problem. Of particular interest in this setting are axioms concerning the delay faced by an agent, which captures the waiting time between being born and getting a slot allocated. We propose three natural rules in this setting and discuss their strengths and weaknesses with respect to delay, incentives and efficiency.

1.1 Introduction

This chapter studies the assignment of agents to slots that are arranged on a line. Imagine a technology that gives out a "slot" per time period, every period, potentially forever. Agents are "born" at some period and want a slot as soon as possible once they are born. Owing to its geometric interpretation, we refer to the problem as the assignment of agents on a line. Slots, before an agent is born, are useless to her. There are waiting costs incurred once the agent is born, and as a result, slots get progressively worse for an agent as one moves further to the right from that agent's birth date. We study this problem in a static deterministic setting, without allowing for transfers or side payments. Agents in our model are completely characterized by their birth dates.

The existing literature studying similar problems usually considers richer preference domains allowing for monetary transfers or randomization, where an agent is not identified by her birth date. Agents differ with respect to waiting costs and/or risk attitudes (or, preferences over fractional assignments). The deterministic setting considerably simplifies the analysis but is still important to study in light of several examples (presented below), where monetary transfers or randomization may not always be appropriate. Our work is also related to the existing literature (with deterministic setting) on the allocation of indivisible objects to agents with unit-demand. These previous works allow for very general preferences, while the focus here is to restrict attention to a smaller preference domain, which nevertheless can be used to model a wide range of situations as described below. The study of such smaller (yet new) domains is important in its own right for the following reason: when one shrinks a preference domain, the class of desirable (e.g., incentive-compatible) mechanisms expands. Therefore, even if the larger domain admits "nice" mechanisms, it becomes important to devise ways to select between already existing mechanisms and new mechanisms that become available in the smaller domain.

The model, albeit quite simple, has applications within a wide range of real-world problems. Consider the scenario where there are a number of large-scale governmental projects to be undertaken. Usually, there is a consensus as to which projects are more desirable, and therefore, we can think of them as being arranged on a line, starting with the biggest or the most desirable project. Firm-specific capacity constraints (stemming from technological, deadline/budget-related, or other relevant considerations) might prevent it from competing for the very top projects. Therefore, capacity constraints determine the birth dates for every firm.

Another important class of examples that fit our framework is the task of assigning users to bottleneck facilities: researchers waiting to use a supercomputer, ships waiting to load/unload cargo at a port, airlines waiting to use the runway for take-off/landing, electric vehicles waiting to be re-charged at a power station, consumers waiting to be picked up by a ride-hailing service, patients waiting for hospital beds and so on.

Another cluster of situations arise in the context of quality line examples where all objects are ordered by quality. Agents prefer better quality, but may feel reluctant to aim above a certain quality level. This is reasonable to assume in many scenarios where agents might be constrained by a budget, or they may feel intimidated aiming too high, or they may feel it is too serious or strong above a certain level. Situations like these arise, for instance, when students consider applying to schools or universities; or, individuals decide to join a sports club or music class; or, parents contemplate putting their children to the best nursery that they can afford; or, citizens make purchasing decisions like buying the largest or best affordable house; firms wish to hire the most skilled worker but are constrained by a hiring budget, and so on.

Clearly, agents in all these scenarios may have different birth dates, and being allocated a slot before an agent is born is not only useless to her but also bad in terms of efficiency. It is not hard to imagine why waiting for a slot after an agent is born is costly in all the above cases.

In these examples, a mechanism design approach is usually called for: a central agency is tasked with designing a mechanism that will systematically assign agents to slots, by meeting a set of desirable goals or axioms; taking into account that agents act in a strategic manner.

1.1.1 Informal discussion of design goals

The most essential welfare determining factor in our model is the *delay* faced by an agent. Delay captures the waiting time between the agent's birth and the slot allocated to the agent. Accordingly, an important goal for the designer is to *minimize the aggregate delay* faced by all agents. Minimizing aggregate delay might come at the expense of a select few agents. Therefore, in light of fairness, an alternative approach might be to *minimize the maximum delay* faced by any agent; or better yet, to minimize lexicographically the vector of delays. A good allocation should also be efficient in the sense of Pareto, i.e., it should not be possible to increase welfare (reduce delay) of an agent without hurting (increasing delay) of another.

The mechanism must take as input the reports of individual birth dates from the agents to be able to propose a "good" allocation. This gives rise to the possibility of agents lying about their birth date to game the system in their favour. Therefore, an important goal for the designer is to construct a mechanism that is immune to manipulations of various kinds. A common starting point is requiring that the mechanism satisfies *strategy-proofness*[1]. It ensures that no individual can get a better slot by lying compared to the one they are assigned by being truthful, irrespective of the reports made by all other agents. Strategy-proofness plays a central role in mechanism design and is frequently imposed as a design requirement in theoretical analyses, across a broad range of assignment, auction, and matching problems.

A strengthening of this notion is *group strategy-proofness* which ensures that no group of the individuals can gain (i.e., each member gets a weakly better slot and at least one member gets a strictly better slot) by jointly lying. Group strategy-proof mechanisms have been analyzed in various settings in the literature: public good provision (Moulin, 1994); house allocation (Pycia and Ünver, 2017; Ehlers, 2002; Pápai, 2000); allocation of a perfectly divisible commodity (Klaus, 2001); matching with contracts (Hatfield and Kojima, 2009).

Group strategy-proofness, although quite strong, still allows for another form of manipulation where: two individuals, by jointly lying, change the outcome of the mechanism, and thereafter, strictly gain (i.e., both agents get strictly better slots) by swapping their allocations ex-post[2]. *Pairwise reallocation-proofness* rules out such forms of manipulations and has been studied in the context of allocation of

[1] For general introduction to the axiom, see Moulin (2014) and Barbera (2001). For various applications of the axiom, see Abdulkadiroğlu and Sönmez (1999), Moulin (1980), Sönmez (1999), Jackson and Nicolo (2004), Klaus et al. (1998), Moulin (2017), Maniquet and Sprumont (1999).

[2] See Moulin (2014) for a discussion of manipulation via swapping objects ex post in the context of the housing market model.

indivisible objects (Pápai, 2000; Mandal and Roy, 2021). A strengthening of this notion to *strongly pairwise reallocation-proofness* ensures that such manipulations do not occur even when the deviating pair of agents only weakly gain, i.e., one agent gets a strictly better slot while the other gets the same slot when they swap their allocated slots (received after jointly lying) ex-post.

Another criterion frequently used in the context of strategy-proof assignment, is the notion of *non-bossiness* which is due to Satterthwaite and Sonnenschein (1981). It ensures that individuals cannot be bossy, that is, change the allocation of others by misreporting without affecting their own. Also, if an allocation violates non-bossiness, then it may invite strategic manipulation: an agent affected by another might pay a small transfer to the latter in return of a false report that results in a preferable allocation to him. As the latter agent is not affected by changing her own report, she may well agree to engage in such manipulations. Normatively, the concept also evokes a form of fairness: it is arguably unfair for an agent to be affected by a change of report made by someone else, even though the change has no consequence on the allocation of the latter.

A designer may also wish to prevent agents from engaging in malicious behaviour. For instance, suppose an agent is assigned a preferred slot of another agent. Now, if the latter, by misreporting, can harm the former, then this agent may wish to do so out of spite. An *envy-proof*[3] mechanism prevents such behaviour: an agent who is envious of another agent's allocated slot, cannot harm the latter by misreporting when the implemented mechanism is envy-proof.

1.1.2 Three Assignment Rules

We propose three assignment rules (or, mechanisms) that naturally arise in our context. We will evaluate their performance with respect to the desirable desiderata discussed in the previous sub-section. In particular, we will see that none of these rules are perfect. They each have their own strengths and weaknesses. This is the

[3] A weaker version of envy-proofness called top-envy-proofness has been studied in the context of allocation of indivisible objects in Mandal and Roy (2021).

subject of discussion in the next sub-section.

1. *Fixed priority rule*: Agents pick a slot in turn following a given priority order (set by the designer) from the set of remaining slots (Definition 1.3.1).
2. *First come first served rule*: A priority order is initially set by the designer after which all agents born in the first period are first assigned (their best available slot) according to that order; next, all agents born in the second period are assigned (their best available slot) according to the priority order; and so on (Definition 1.3.2).
3. *Slot-based priority rule*: This rule is a generalization of the fixed priority rule where every slot is endowed with a priority order (possibly different) over agents. Among all agents born in the first period, the agent with the highest priority for the slot gets it; next among all agents who are either born in the second period or born earlier but are still unassigned, the one with the highest priority for the second slot gets it; and so on (Definition 1.3.3).

1.1.3 Brief summary of results

We examine the compatibility of the different goals with each other in our model. In particular, we report which combinations of these properties are feasible concurrently as well as those that are not.

In our model, Pareto efficiency happens to be equivalent to minimizing aggregate delay (Proposition 1.4.1). We also show that group strategy-proofness is equivalent to the combination of individual strategy-proofness and nonbossiness (Proposition 1.4.2).

The attractive incentive compatibility property of strategy-proofness is in direct conflict with the two main delay concerns: minimizing maximum delay and leximax minimizing the vector of delays (Proposition 1.4.3). If one drops these two properties and only insists on mechanisms that minimize aggregate delay, the question remains whether it is possible to avert manipulations of different kinds as discussed earlier. We prove a negative result: the impossibility of Pareto efficient, group strategy-proof, strongly pairwise reallocation-proof and envy-proof mechanisms when there are at least three agents (Theorem 1.4.1). This impossibility result uses the stronger counterpart of every property wherever applicable. Weakening strongly pairwise reallocation-proofness to pairwise reallocation-proofness yields a positive result. Indeed, the fixed priority rule satisfy all of them (Proposition 1.4.4).

Since the fixed priority rule is strategy-proof, we know for sure that it does not minimize max delay. But it is worthwhile to quantify in some way how poorly it fares in minimizing max delay. We take the worse case approach to determine this (see, Definition 1.4.2). The worst-case analysis reveals that any Pareto efficient, strategy-proof, and nonbossy assignment rule is the worst rule with respect to minimizing max delay in our domain (Proposition 1.4.5). As a corollary, we then obtain that the fixed priority rule is therefore one of the worst assignment rules with respect to minimizing max delay (Corollary 1.4.2).

The first come first served rule fares much better in this regard: it leximax minimizes delays (Proposition 1.4.6), but is not immune to strategic manipulations of any kind (Proposition 1.4.7).

The slot-based priority rule is also Pareto efficient and group strategy-proof, and offers a little more flexibility over the fixed priority rule when the designer has to respect exogenously given slot priorities (Proposition 1.4.8). However, it fails to be pairwise reallocation proof and envy-proof (Proposition 1.4.10).

1.1.4 Related literature

Our work contributes to the recent literature on slot allocation problems: a group of agents must be assigned to a slot located along a line. The closest to our model is the paper by Hougaard et al. (2014). They study a problem where each agent has a preferred slot which is called the target, and wants to be served as close as possible to it. Preferences are single-peaked and symmetric (to both sides of the peak). Each slot can serve only one agent. They first consider a deterministic assignment of agents to slots and provide a direct method for testing if a given deterministic assignment is aggregate gap-minimizing. Next, they consider probabilistic assignment of agents to slots and propose an aggregate gap minimizing modification of the random priority method to solve the problem. Our domain in a sub-domain of their domain.

Chun and Park (2017) studies a cardinal version of the above problem. They consider a setting where agents differ in their most preferred slot position, which they call the peak. Agents have quasi-linear preferences over slots and money: each agent's utility from her assignment is equal to the amount of monetary transfer minus the distance between the peak and her assigned slot. They propose two assignment rules as a solution to their problem: the leximin and the leximax rules. While we also assume that the (per period) waiting cost is identical across agents like these papers, our setting has an important difference: both these papers allow agents to be assigned a slot either before or after their peak (or, target), while in our model, agents have birth dates and can only be assigned a slot after they are born.

Crès and Moulin (2001) and Bogomolnaia and Moulin (2002) study scheduling problems with deadlines, using randomization as a tool to restore (ex ante) fairness. Agents in their model arrive in batches and therefore have the same birth date but are heterogeneous in their "deadline", i.e., the number of time periods they can afford to wait for service.

Ghosh et al. (2020) study a model similar in flavour to ours in a cardinal environment. Agents from a finite population arrive at various discrete times and exit after they use a server for one period each. Each agent has a per-period cost of queuing, which constitutes his private information. Agents in their model are strategic about their waiting cost but not their date of arrival. On the other hand, birth date report is a strategic variable in our setting.

Assignment of heterogeneous indivisible goods with general single-peaked preferences where each individual can receive at most one good have been studied by Bade (2019) and more recently by Mandal and Roy (2021).

1.1.5 Outline of the chapter

Section 1.2 presents the model and discusses the properties and structure of allocations in our setting. Section 1.3 presents formal definitions of the three assignment rules along with definitions of the desirable goals (informally discussed in the introduction) the designer seeks to achieve. Section 1.4 presents the results and some discussion. We end with concluding remarks highlighting the implications of our results. We also discuss some potential new avenues to explore for further research.

1.2 Model

Slots are arranged on a line and therefore the set of slots can be identified by the set of integers. Agents have a most preferred slot (birthdate) on this line and want a slot sooner rather than later once they are born. Henceforth, we refer to the most preferred slot of an agent as her *target*.

A *problem* is represented by a triple $\langle N,\ S,\ t_N \rangle$ consisting of:

* A set of agents $N = \{1,2,3,\cdots,|N|\}$.
* A set of slots $S = \{1,2,3,...\}$.
* A *profile of targets* $t_N = (t_i)_{i \in N}$, where $t_i \in S$ is the target slot of agent $i \in N$.

The set of agents N is fixed for the rest of the paper.

We use the phrases “profile” and “profile of targets” interchangeably. For any group of agents $T \subseteq N$, the list of targets of agents in T is denoted as $t_T = (t_i)_{i \in T} \in S^{|T|}$. Following the standard practise in the literature, we sometimes denote a profile as $t_N = (t_i, t_{-i})$ or, as $t_N = (t_T, t_{-T})$ where $t_{-T} = (t_i)_{i \in N \setminus T} \in S^{|N \setminus T|}$.

Allocations

Allocations specify the slot that every agent receives. Formally, an *allocation* is a map $\alpha : N \to S$ such that $i \neq j \implies \alpha(i) \neq \alpha(j)$; i.e., no two agents are assigned to the same slot. $\alpha(i)$ is called the allocation of agent i at α. For notational simplicity, we sometimes denote $\alpha(i)$ by α_i if there is no risk of confusion. All possible allocations are collected in the set $\mathscr{A}$.

Delay experienced by agents

Delay is the primary welfare determining factor in our model. Delay captures the waiting time between the agent's birth and the slot allocated to the agent, i.e., the distance between the agent's allocated slot and her target.

For any *allocation* $\alpha \in \mathscr{A}$, the *delay* suffered by any agent i with target t_i in allocation α, denoted by $\delta_i(\alpha, t_i)$, is defined as,

$$\delta_i(\alpha, t_i) \;:=\; \alpha(i) \,-\, t_i$$

Preferences of agents

Preference of agents reflect the fact that an *earlier slot is better* all else being constant. For each agent $i \in N$, her target slot $t_i \in S$ induces a weak preference relation $R_i^{t_i}$ (with strict and indifference parts $P_i^{t_i}$ and $I_i^{t_i}$ respectively) over the set of allocations $\mathscr{A}$ as follows. For all allocations $\alpha, \alpha' \in \mathscr{A}$:

- if $\delta_i(\alpha, t_i) \geq 0$ and $\delta_i(\alpha', t_i) \geq 0$, then:
  * $\alpha_i \,<\, \alpha'_i \implies \alpha\, P_i^{t_i}\, \alpha'$
  * $\alpha_i \,=\, \alpha'_i \implies \alpha\, I_i^{t_i}\, \alpha'$
- if $\delta_i(\alpha, t_i) \geq 0$ and $\delta_i(\alpha', t_i) < 0$, then $\alpha\, P_i^{t_i}\, \alpha'$
- if $\delta_i(\alpha, t_i) < 0$ and $\delta_i(\alpha', t_i) < 0$, then $\alpha\, I_i^{t_i}\, \alpha'$

A profile of targets $t_N = (t_i)_{i \in N}$, therefore, induces a profile of preference relations $R^{t_N} = (R_i^{t_i})_{i \in N}$.

1.2.1 Desirable properties of allocations

In this subsection, we define and discuss several properties that we would like allocations to satisfy.

The first one is called *individual rationality*. It says that no individual should be allocated a slot earlier than her target since she has no use for it.

Definition 1.2.1. *Given a profile of targets* $t_N = (t_i)_{i \in N}$*, an allocation* $\alpha \in \mathscr{A}$ *is* ***individually rational*** *if for every agent* $i \in N$ *we have* $\delta_i(\alpha,\, t_i) \geq 0$

Since delays faced by individuals are costly, a good allocation should seek to allocate slots in a manner so that, the delay that agents face is as little as possible. There are several different ways of thinking about doing so. The designer may want to minimize the total delay faced by all individuals cumulatively. Alternatively, he may instead pursue a goal of minimizing the maximum delay that any single agent faces. Yet another goal and a stronger one, might be to lexicographically minimize the vector of delays. These are formally presented in the following four definitions.

Definition 1.2.2. *Given a profile of targets* $t_N = (t_i)_{i \in N}$*, an allocation* $\alpha \in \mathscr{A}$ *is said to* ***minimize aggregate delay*** *if it is individually rational and for any other individually rational allocation* $\alpha' \in \mathscr{A}$ *we have,*

$$\sum_{i \in N} \delta_i(\alpha,\, t_i) \;\leq\; \sum_{i \in N} \delta_i(\alpha',\, t_i)$$

Definition 1.2.3. *Given a profile of targets* $t_N = (t_i)_{i \in N}$*, an allocation* $\alpha \in \mathscr{A}$ *is said to* ***minimize max delay*** *if it is individually rational and for any other individually rational allocation* $\alpha' \in \mathscr{A}$ *we have,*

$$\max\left\{\delta_i(\alpha,\, t_i) \;\middle|\; i \in N\right\} \;\leq\; \max\left\{\delta_i(\alpha',\, t_i) \;\middle|\; i \in N\right\}$$

Definition 1.2.4. *The leximax order* $\prec_{leximax}$ *on* $\mathbb{R}^n$ *is defined as follows. For any* $x = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$*, let* $x^{\downarrow} = (x_1^{\downarrow}, x_2^{\downarrow}, \cdots, x_n^{\downarrow}) \in \mathbb{R}^n$ *be a permutation of* *the coordinates of x in non-increasing order:* $x_1^{\downarrow} \geq x_2^{\downarrow} \geq \cdots \geq x_n^{\downarrow}$. *We say that the vector* $x \prec_{leximax} y$ *if there is a* $j \in \{1, 2, \cdots, n\}$ *such that* $x_j^{\downarrow} < y_j^{\downarrow}$, *while* $x_i^{\downarrow} = y_i^{\downarrow}$ *for all* $i < j$.

Definition 1.2.5. *Given a profile of targets* $t_N = (t_i)_{i \in N}$ *and an allocation* $\alpha \in \mathscr{A}$, *we denote by* $\delta^{\alpha}(t_N) = \big(\delta_i(\alpha,\, t_i)\big)_{i \in N}$ *the vector of delays.*
Then, an allocation $\alpha \in \mathscr{A}$ *is said to* ***leximax minimize delays*** *if it is individually rational and for any other individually rational allocation* $\alpha' \in \mathscr{A}$ *we have,*

$$\delta^{\alpha}(t_N) \ \prec_{leximax} \ \delta^{\alpha'}(t_N)$$

Pareto Efficiency of an allocation ensures that it cannot be improved upon in the sense of Pareto (that is, every agent is made weakly better off and at least one agent is made strictly better off).

Definition 1.2.6. *An allocation* $\alpha \in \mathscr{A}$ *is* ***Pareto dominated*** *by another allocation* $\alpha' \in \mathscr{A}$ *at a profile of targets* t_N, *if* $\alpha' R_i^{t_i} \alpha \quad \forall i \in N$, *and* $\alpha' P_j^{t_j} \alpha$ *for at least one agent* $j \in N$. *An allocation* $\alpha \in \mathscr{A}$ *is said to be* ***Pareto efficient*** *at a profile* t_N *if it is not Pareto dominated by any other allocation at that profile.*

We now record a few lemmas that will be useful in the ensuing analysis. They are proved in Appendix A.

Lemma 1.2.1. *An allocation* α *is Pareto efficient at the profile of targets* t_N *if and only if there does not exist an unassigned slot* $s \in S$ *and an agent* i *with target* $t_i \leq s$ *such that* $\alpha(i) > s$.

An immediate consequence of Lemma 1.2.1 is the simple structure of Pareto efficient allocations. All Pareto efficient allocations are a split between “intervals” of slots: (1) there is at least one agent who desires the first slot of the interval,

(2) all slots in the interval are assigned, (3) all agents with targets slots within the interval are assigned a slot on the interval. We formalize this in the next lemma.

Lemma 1.2.2. *For any profile of targets $t_N = (t_i)_{i \in N}$, the set of slots S can be uniquely partitioned into $\{S_0, S_1, S_2, \cdots\}$ such that for any Pareto efficient allocation α we have:*

- $S_0 = \emptyset \iff$ *there is at least one agent i with target $t_i = 1$.*
- *If $s \in S_k$ and $s' \in S_{k+1}$ then we have $s < s'$.*
- *A slot s belongs to S_k for k even if and only if it is unassigned:* $\alpha^{-1}(s) = \emptyset \iff s \in S_{2k}$ *for some $k \in \{0, 1, 2, \cdots\}$.*
- *A slot s belongs to S_k for k odd if and only if it is assigned:* $\alpha^{-1}(s) \neq \emptyset \iff s \in S_{2k+1}$ *for some $k \in \{0, 1, 2, \cdots\}$.*
- *For every k odd, there is an agent i_k^* such that her target $t_{i_k^*} = \min S_k$.*
- *The target slot of agent i belongs to S_k if and only if agent i is assigned a slot in S_k, i.e.,* $t_i \in S_k \iff \alpha_i \in S_k$

In our environment, Pareto efficiency is equivalent to minimizing aggregate delay.

Lemma 1.2.3. *An allocation is Pareto efficient if and only if it minimizes aggregate delay.*

Lemma 1.2.4. *Let α, $\alpha' \in \mathscr{A}$ be Pareto efficient allocations at profile t_N. Then,*

$$\sum_{i \in N} \delta_i(\alpha, t_i) = \sum_{i \in N} \delta_i(\alpha', t_i)$$

Proof. Follows immediately from Lemma 1.2.3 ■

Lemma 1.2.5. *Let α be a Pareto efficient allocation at the profile of targets t_N such that assigned slots are ordered like targets (i.e., $t_i < t_j$ implies $\alpha_i < \alpha_j$ for each pair of agents i, j). Then, α leximax minimizes delays at t_N.*

1.3 Assignment Rules

Recall, the set of all possible *allocations* is denoted by $\mathscr{A}$.

An **assignment rule (or, mechanism)** is a function $\Psi : S^{|N|} \longrightarrow \mathscr{A}$ that systematically chooses an allocation $\Psi(t_N) \in \mathscr{A}$ for each profile of reported targets $t_N \in S^{|N|}$. For every agent $i \in N$, let $\Psi_i(t_N)$ denote the allocation (allocated slot) of agent i at the reported target profile t_N according to the allocation rule Ψ.

1.3.1 Natural assignment rules in the context

In this subsection, we propose examples of assignment rules that naturally arise in the environment that we study in this article.

Let the *best slot* among a set of slots $Y \subseteq S$ given a target slot t_i be denoted as,

$$B\big(t_i,\, Y\big) \,:=\, \min\{\, s' \in Y \mid s' \geq t_i\,\}$$

Definition 1.3.1. *The **fixed priority rule** is parameterized by an arbitrary priority order over the set of agents N, which is fixed a priori. A **priority** is a bijection $\sigma : N \to N$, i.e., an ordering over the set of agents. For any fixed priority σ, the fixed priority rule ψ_σ^{FP} is defined inductively. For a given profile of targets $t_N = (t_i)_{i \in N}$, construct an allocation α as follows:*

$$\begin{aligned}
\alpha(\sigma(1)) &= B\big(t_{\sigma(1)},\, S\big)\\
\alpha(\sigma(2)) &= B\big(t_{\sigma(2)},\, S \setminus \{\alpha(\sigma(1))\}\big)\\
\alpha(\sigma(3)) &= B\big(t_{\sigma(3)},\, S \setminus \{\alpha(\sigma(1)), \alpha(\sigma(2))\}\big)\\
&\vdots\\
\alpha(\sigma(|N|)) &= B\big(t_{\sigma(|N|)},\, S \setminus \{\alpha(\sigma(1)), \alpha(\sigma(2)), \cdots, \alpha(\sigma(|N|-1))\}\big)
\end{aligned}$$

The fixed priority rule assigns $\psi_\sigma^{FP}(t_N) \,=\, \alpha$.

Definition 1.3.2. *The **first come first served rule** is parameterized by an arbitrary priority order over the set of agents N, which is fixed a priori. A priority is a* *bijection $\sigma : N \to N$, i.e., an ordering over the set of agents. For any fixed priority σ, the first come first served rule ψ^{FCFS}_{σ} computes an allocation in two steps. For any given profile of targets $t_N = (t_i)_{i\in N}$,*

(i) *Derive the **first come first priority** $\sigma_{FF} : N \to N$ from σ as follows: For any two agents i and j,*

   * *if* $t_i < t_j$ *then* $\sigma^{-1}_{FF}(i) < \sigma^{-1}_{FF}(j)$
   * *if* $t_i = t_j$ *then* $\sigma^{-1}_{FF}(i) < \sigma^{-1}_{FF}(j) \iff \sigma^{-1}(i) < \sigma^{-1}(j)$

(ii) *Then, the first come first served rule assigns* $\psi^{FCFS}_{\sigma}(t_N) = \psi^{FP}_{\sigma_{FF}}(t_N)$.

Definition 1.3.3. *The **slot-based priority rule** is parameterized by an arbitrary profile of priority orders over the set of agents N, which is fixed a priori. For any slot $s \in S$, a priority order for slot s, is simply a linear order $\gg_s$ over N. A **profile of priority orders** is a list $\gg = (\gg_s)_{s\in S}$.*

Fixing a profile of priority orders $\gg = (\gg_s)_{s\in S}$, the slot-based priority rule with respect to $\gg$ associates with every profile of targets $t_N = (t_i)_{i\in N}$ an allocation α which is constructed as follows:

- ***Step 0 :** For all $k \in S$, define $I^k = \{i \in N \mid t_i \leq k\}$*
- ***Step 1 :** If $I^1 \neq \emptyset$, then select the agent $i^1 \in I^1$ who has the highest priority according to $\gg_1$. Assign $\alpha(i^1) = 1$. Else,* slot 1 *remains unassigned and $i^1 = \emptyset$.*
- ***Step k = 2, 3, ... :** If $I^k \setminus \bigcup_{j=1}^{k-1} i^j \neq \emptyset$, select the agent $i^k \in I^k \setminus \bigcup_{j=1}^{k-1} i^j$ who has the highest priority according to $\gg_k$. Assign $\alpha(i^k) = k$. Else,* slot *k is unassigned and $i^k = \emptyset$. Continue this procedure until all agents are allocated.*

Then, the slot-based priority rule assigns $\psi^{SBP}_{\gg}(t_N) = \alpha$

Remark 1.3.1. *It is easy to see that the slot-based priority rule is output equivalent to the celebrated agent proposing DA algorithm of Gale and Shapley (1962). We* *present it in the manner above because it seems to be the most natural way to do it in our restricted domain.*

Remark 1.3.2. *When the priority order of every slot is identical, then the slot-based priority rule outputs the same allocation as the fixed priority rule.*

1.3.2 Desirable properties of Assignment Rules

In this subsection, we define and discuss several desirable properties with respect to which an assignment rule may be evaluated.

The first three properties are specific to the context under study in this article. They focus on the *delay* faced by agents in the allocation proposed by the assignment rule, at every profile of targets reported by the agents.

Definition 1.3.4. *An assignment rule Ψ* ***minimizes aggregate delay*** *if, for every profile $t_N \in S^{|N|}$, the allocation $\Psi(t_N)$ miminizes aggregate delay.*

Definition 1.3.5. *An assignment rule Ψ* ***minimizes max delay*** *if, for every profile $t_N \in S^{|N|}$, the allocation $\Psi(t_N)$ miminizes max delay.*

Definition 1.3.6. *An assignment rule Ψ* ***leximax minimizes delays*** *if, for every profile $t_N \in S^{|N|}$, the allocation $\Psi(t_N)$ leximax minimize delays.*

The following properties are more universal in nature and apply to more general environments than the ones studied in this article.

An assignment rule is *Pareto efficient* if it always outputs a Pareto efficient allocation at every reported profile of targets.

Definition 1.3.7. *An assignment rule Ψ is* ***Pareto efficient*** *if, for every profile $t_N \in S^{|N|}$, the allocation $\Psi(t_N)$ is Pareto efficient.*

Next, we define three *non-cooperative* properties of assignment rules. They are non-cooperative in the sense that they relate an assignment rule's outcome under two scenarios, when a single agent makes unilateral target (preference) revelation deviations.

The first such property is called *strategy-proofness*. It ensures that no individual can gain by reporting false targets (preferences).

Definition 1.3.8. *An agent i **manipulates** an assignment rule Ψ at a profile of targets $t_N = (t_i, t_{-i}) \in S^{|N|}$ by reporting $t_i' \in S$ if*

$$\Psi\big(t_i',\, t_{-i}\big)\ P_i^{t_i}\ \Psi\big(t_i, t_{-i}\big)$$
$$\text{i.e.,}\quad t_i \ \le\ \Psi_i\big(t_i',\, t_{-i}\big)\ <\ \Psi_i\big(t_i, t_{-i}\big)$$

*An assignment rule Ψ is **strategy-proof** if it is never manipulated by any individual agent.*

A strategy-proof assignment rule is appealing because it gives straightforward incentives to each individual participant, whether or not this particular agent knows anything more than her own targets (preferences): any information about other participants' targets (preferences) is useless to her, as long as she cannot coordinate his actions with that of any other agent.

The following axiom is called *non-bossiness* and is due to Satterthwaite and Sonnenschein (1981). It ensures that individuals cannot be bossy, that is, change the allocation for others, by reporting different targets, without changing their own.

Definition 1.3.9. *An assignment rule Ψ is **non-bossy** if for all profile $t_N \in S^{|N|}$ and for all agent $i \in N$ and for all $t_i' \in S$,*

$$\Psi_i\big(t_i,\, t_{-i}\big) = \Psi_i\big(t_i',\, t_{-i}\big) \quad \Longrightarrow \quad \Psi\big(t_i,\, t_{-i}\big) = \Psi\big(t_i',\, t_{-i}\big)$$

Next, we define the notion of *envy-proofness*[4] of an assignment rule. This property requires that if an individual j is assigned a preferred slot of another agent i, then no matter how agent i misreports her target, agent j cannot be worse-off. In other words, an agent who is envious of another agent's allocated slot, cannot harm her by misreporting.

Definition 1.3.10. *An assignment rule Ψ is **envy-proof** if for all profiles of targets $t_N = (t_i)_{i \in N}$ and all distinct agents i, $j \in N$,*

$$\left[\, t_i \leq \Psi_j(t_N) \;<\; \Psi_i(t_N) \,\right] \quad \Longrightarrow \quad \left[\, t_j \leq \Psi_j(t_i',\, t_{-i}) \;\leq\; \Psi_j(t_N) \,\right] \forall t_i' \;\in\; S$$

The following properties are *co-operative* in nature: they relate an assignment rule's outcomes under two scenarios when a group or coalition of agents jointly make target (preference) revelation deviations.

The first such notion that is defined below is called *group strategy-proofness* of an assignment rule. It is a stricter requirement than strategy-proofness. It ensures that no subset or group of individuals can gain by reporting false targets. More precisely, by colluding and jointly misreporting targets, no individual among the deviators can be made better off without hurting at least one other deviator.

Definition 1.3.11. *A group of agents $T \subseteq N$ **jointly manipulates** an assignment rule Ψ at profile $t_N = (t_T, t_{-T}) \in S^{|N|}$ by jointly reporting $t_T' \in S^{|T|}$ if,*

$$t_i \leq \; \Psi_i\big(t_T',\, t_{-T}\big) \leq \; \Psi_i\big(t_T, t_{-T}\big) \quad \forall i \in T$$
$$t_j \leq \; \Psi_j\big(t_T',\, t_{-T}\big) < \; \Psi_j\big(t_T, t_{-T}\big) \quad \textit{for at least one agent } j \in T$$

*An assignment rule Ψ is **group strategy-proof** if it is never jointly manipulated by any group of agents.*

[4]A weaker version of envy-proofness called top-envy-proofness has been studied in the context of allocation of indivisible objects in Mandal and Roy (2021).

Next, we present the notion of *strongly pairwise reallocation-proof* assignment rules. It requires that no pair of agents can misreport their targets and be jointly better-off redistributing their allocations ex-post. The pair of agents is said to be jointly better-off if each member is weakly better-off and at least one member is strictly better-off.

Definition 1.3.12. *An assignment rule Ψ is* ***weakly manipulable through pairwise reallocation*** *if there exists a profile of targets $t_N = (t_i, t_j, t_{-\{i,j\}}) \in S^{|N|}$, distinct agents $i,\ j \in N$, and $t'_i,\ t'_j \in S$ such that*

(a) $t_i \leq \Psi_j(t'_i,\ t'_j,\ t_{-\{i,j\}}) \leq \Psi_i(t_N)$

(b) $t_j \leq \Psi_i(t'_i,\ t'_j,\ t_{-\{i,j\}}) < \Psi_j(t_N)$

An assignment rule Ψ is ***strongly pairwise reallocation-proof*** *if it is not weakly manipulable through pairwise reallocation.*

The next property is a weakening of the notion of strongly pairwise reallocation-proofness to *pairwise reallocation-proofness*. This property requires that no pair of agents can misreport their targets and be strictly better-off redistributing their allocations ex-post. Unlike in the stronger counterpart, a pair of agents now manipulates only if it leads to a strictly better outcome for both agents after trading allocations ex-post.

Definition 1.3.13. *An assignment rule Ψ is* ***strongly manipulable through pairwise reallocation*** *if there exists a profile of targets $t_N = (t_i, t_j, t_{-\{i,j\}}) \in S^{|N|}$, distinct agents $i,\ j \in N$, and $t'_i,\ t'_j \in S$ such that*

(a) $t_i \leq \Psi_j(t'_i,\ t'_j,\ t_{-\{i,j\}}) < \Psi_i(t_N)$

(b) $t_j \leq \Psi_i(t'_i,\ t'_j,\ t_{-\{i,j\}}) < \Psi_j(t_N)$

An assignment rule Ψ is ***pairwise reallocation-proof*** *if it is not strongly manipulable through pairwise reallocation.*

Remark 1.3.3. *Note that, if an agent is allocated her target at some profile, then she does not wish to manipulate with another agent through pairwise reallocation.*

1.4 Results

In this section, we present results having two flavours. The first type tries to address the (in)compatibility of the different axioms with each other. The other kind has to do with evaluating the proposed assignment rules with respect to the desirable desiderata presented in the previous section.

An immediate observation that follows from Lemma 1.2.3 gives us the following proposition.

Proposition 1.4.1. *An assignment rule is Pareto efficient if and only if it minimizes aggregate delay.*

In the current environment, it turns out that group strategy-proofness is equivalent to the conjunction of two non-cooperative properties: strategy-proofness and non-bossiness.

Proposition 1.4.2. *An assignment rule is group strategy-proof if and only if it is strategy-proof and nonbossy.*

Proof. **(If part)** Suppose Ψ is strategy-proof and nonbossy. Assume for contradiction that Ψ is not group strategy-proof. Since Ψ is not group strategy-proof, there exists a profile $t_N \in S^{|N|}$, a group of agents $T \subseteq N$ and $t'_T \in S^{|T|}$ such that

$$\begin{aligned} &t_i \leq \Psi_i\big(t'_T, t_{-T}\big) \leq \Psi_i\big(t_T, t_{-T}\big) \quad \forall i \in T \\ &t_j \leq \Psi_j\big(t'_T, t_{-T}\big) < \Psi_j\big(t_T, t_{-T}\big) \quad \text{for at least one agent } j \in T \end{aligned}$$

Let $T = \{1, 2, \cdots p\}$. For all $i \in T$, let $s^*_i = \Psi_i\big(t'_T, t_{-T}\big)$. Now, $s^*_i \leq \Psi_i\big(t_N\big) \ \ \forall i \in T$. Next consider another target $\tilde{t}_i$ of agent $i \in T$ such that $\tilde{t}_i = s^*_i$. Strategy-proofness of Ψ then implies $\Psi_1\big(\tilde{t}_1,\, t_{-1}\big) = \Psi_1\big(t_N\big)$. Next by non-bossiness, $\Psi\big(\tilde{t}_1,\, t_{-1}\big) = \Psi\big(t_N\big)$. Repeating the same argument for agents $2,3,\cdots p$, we get

$$\Psi\big(\tilde{t}_T,\, t_{-T}\big) = \Psi\big(t_N\big)$$

Next, consider the profile $(t'_T,\, t_{-T})$. Fix agent $j \in T$ and consider deviation $\tilde{t}_j$ at this profile. Since $\tilde{t}_j = s^*_j = \Psi_j\big(t'_T,\, t_{-T}\big)$, by strategy-proofness $\Psi_j\big(\tilde{t}_j,\, t'_{T\setminus\{j\}},\, t_{-T}\big)$ $= \Psi_j\big(t'_T,\, t_{-T}\big)$. Non-bossiness then implies $\Psi\big(\tilde{t}_j,\, t'_{T\setminus\{j\}},\, t_{-T}\big) = \Psi\big(t'_T,\, t_{-T}\big)$. Continuing in this manner, we can move targets of all agents $j \in T$ from t'_j to $\tilde{t}_j$ one by one and obtain,

$$\Psi\big(\tilde{t}_T,\, t_{-T}\big) = \Psi\big(t'_T,\, t_{-T}\big)$$

,which then implies $\Psi\big(t'_T,\, t_{-T}\big) = \Psi\big(t_N\big)$, a contradiction to the fact that $t_j \leq \Psi_j\big(t'_T,\, t_{-T}\big) < \Psi_j\big(t_T, t_{-T}\big)$ for at least one agent $j \in T$.

(Only-if part) It is straight forward to see that group strategy-proofness implies strategy-proofness and non-bossiness. ■

The next proposition highlights the conflict between strategy-proofness and individual delay concerns.

Proposition 1.4.3. *There does not exist a strategy-proof assignment rule that always*

(a) minimizes max delay.

(b) leximax minimizes delay.

Proof. Consider a problem with three agents $N = \{1,2,3\}$ with target $t_i = 1$ for all $i \in N$. Without loss of generality, suppose a rule assigns allocation α at this profile t where $\alpha(1) = 1$, $\alpha(2) = 2$ and $\alpha(3) = 3$. By strategy-proofness, when agent 2 reports $t'_2 = 2$ while other reports are unchanged, the rule must assign agent 2 again to slot 2, and either agent 1 or agent 3 must be assigned slot 3. Without loss of generality, say at profile $t' = (t_1, t'_2, t_3)$ the rule outputs allocation α' such that $\alpha'(1) = 1$, $\alpha'(2) = 2$ and $\alpha'(3) = 3$. The vector of delays for allocation α' is $(0,0,2)$. But allocation α'' such that $\alpha''(1) = 1$, $\alpha''(2) = 3$ and $\alpha''(3) = 2$ with vector of delay $(0,1,1) \prec_{leximax} (0,0,2)$ and also minimize max delay. ■

If we omit minimizing max delay concerns, but insist on all the other properties (the stronger counterparts wherever applicable), it turns out that, this is not possible in our domain, as is shown in the next impossibility result.

Theorem 1.4.1. *Suppose $|N| \geq 3$. Then, there does not exist a Pareto efficient, group strategy-proof, strongly pairwise reallocation-proof and envy-proof assignment rule.*

Proof. Suppose $N = \{1,2,3\}$ be the set of agents and $S = \{a,b,c,d,e\}$ be the set of slots where a is the earliest slot followed by b being the next and so on. Assume for contradiction that Ψ is a Pareto efficient, group strategy-proof, strongly pairwise reallocation-proof, and envy-proof assignment rule. We adopt the following simplified notation for the rest of the proof. A profile $t_N = (t_1,t_2,t_3)$ is denoted simply as $t_1t_2t_3$ and an allocation $\Psi(t_N)$ at profile t_N where $\Psi_1(t_N) = a$, $\Psi_2(t_N) = b$ and $\Psi_3(t_N) = c$ is denoted as $\Psi(t_N) = abc$.

Next, consider the target profile aaa. Without loss of generality, let us assume $\Psi(aaa) = abc$. Group strategy-proofness then implies,

$$\Psi(t_N) = abc \quad \forall t_N \in \{aab, aac, aba, abb\} \tag{1.1}$$

Then by strongly pairwise reallocation-proofness of Ψ, agent 1 and 2 must get slots a and b between them. Therefore, we must then have

$$\Psi(baa) = bac \tag{1.2}$$

Group strategy-proofness then implies,

$$\Psi(bab) = bac \tag{1.3}$$

Next by Pareto efficiency, $\Psi(bba) \in \{bca, cba\}$. Suppose $\Psi(bba) = cba$. Then at profile baa, agent 2 can misreport to b and swap allocations with agent 3 ex-post and together do weakly better (agent 3 does strictly better). This is a contradiction to strongly pairwise reallocation-proofness of Ψ. Consequently,

$$\Psi(bba) = bca \tag{1.4}$$

Envy-proofness of Ψ together with (1.1), (1.3) and (1.4) then implies neither agent 2 nor agent 3 can get slot b and as a result, by Pareto efficiency we must have,

$$\Psi(bbb) \in \{bcd, bdc\} \tag{1.5}$$

Case:1 Suppose, $\Psi(bbb) = bcd$.

Then by group strategy-proofness of Ψ we get,

$$\Psi(t_N) = bcd \quad \forall t_N \in \{bbc, bcb, bcc\} \tag{1.6}$$

Strongly pairwise reallocation proofness implies agent 1 and 2 must get between them slots b and c. Thus, $\Psi(cbb) = cbd$. Next again by strongly pairwise reallocation proofness, this time between agent 2 and 3 we have,

$$\Psi(ccb) = cdb \tag{1.7}$$

Again, by strongly pairwise reallocation proofness and the fact that $\Psi(cbb) = cbd$ we have,

$$\Psi(cbc) = cbd \tag{1.8}$$

Envy-proofness of Ψ together with (1.6), (1.8) and (1.7) then implies that agent 1 must get the slot c at the profile ccc. Thus,

$$\Psi(ccc) \in \{cde, ced\} \tag{1.9}$$

Then by envy-proofness and Pareto efficiency we get,

$$\Psi(cac) = cad \tag{1.10}$$

$$\Psi(cca) = cda \tag{1.11}$$

Now, strongly pairwise reallocation-proofness together with (1.10) and (1.11) implies that

$$\Psi(acc) \notin \{acd, adc\} \tag{1.12}$$

This is a contradiction to the fact that Ψ is Pareto efficient.

The other case when $\Psi(bbb) = bdc$ can be similarly shown to lead to a contradiction. ■

Weakening *strongly pairwise reallocation-proofness* to *pairwise reallocation-proofness* yields a positive result, as outlined in the next proposition.

Proposition 1.4.4. *Every fixed priority rule is group strategy-proof, Pareto efficient, envy-proof and pairwise reallocation-proof.*

Proposition 1.4.4 is proved in Appendix A.

Although the fixed priority rule satisfies a number of important properties, it has a notable drawback when it comes to max delay concerns. The next remark highlights this with an example.

Remark 1.4.1. *The fixed priority rule can perform very poorly with respect to minimizing max delay.*

To see this, consider the following example with $N = \{1, 2, \cdots, n\}$. Fix priority σ with $\sigma(i) = i$ for all i. Consider the profile t_N where, $t_1 = t_n = 1$ and $t_j = j$ for all $j \in \{2, 3, \cdots, n-1\}$. Then $\psi_\sigma^{FP}(t_N) = \alpha$ where $\alpha(n) = n$ and her delay is $n-1$. Consider instead the allocation α' where $\alpha'(1) = 1$, $\alpha'(n) = 2$ and $\alpha'(j) = j+1$ for all $j \in \{2, 3, \cdots, n-1\}$. The max delay suffered by an agent in α' is 1.

The unattractive feature highlighted in the above remark is not unique to the fixed priority rule. It is unfortunately true for any Pareto efficient, strategy-proof, and non-bossy assignment rule in a sense that we make precise below.

We measure how bad an assignment rule performs with respect to minimizing max delay by the *worst-case scenario*. We will need the notion of *regret* of an assignment rule. To formally define it we will need additional notations.

Recall that, for any profile $t_N = (t_i)_{i\in N}$, we denote by $\delta^{\alpha}(t_N) := \big(\delta_i(\alpha,\, t_i)\big)_{i\in N}$ the vector of delays at that profile corresponding to any allocation α.

For any profile t_N and any allocation α, let

$$\max_{i\in N} \delta^{\alpha}(t_N) := \max\big\{\delta_1(\alpha,\, t_1), \cdots, \delta_n(\alpha,\, t_n)\big\}.$$

Let $\mathscr{A}^{IR}(t_N)$ denote the set of all *individually rational* allocations at profile t_N.

Definition 1.4.1. *The regret corresponding to the assignment rule* Ψ *at any profile of targets* $t_N = (t_i)_{i\in N}$ *is given by*

$$Regret\big(\Psi, t_N\big) := \frac{\max_{i\in N} \delta^{\Psi(t_N)}(t_N)}{\min_{\alpha\in\mathscr{A}^{IR}(t_N)} \; \max_{i\in N} \delta^{\alpha}(t_N)} \tag{1.13}$$

The *regret* of an assignment rule Ψ at a profile of targets t_N is defined as: the ratio between the max delay corresponding to the allocation $\Psi(t_N)$ and the minimum max delay that is feasible at profile t_N.

Next, let $S^* = \{t_N \in S^{|N|} \mid i \neq j \iff t_i \neq t_j\}$ denote the set of target profiles such that no two agents have the same target. Note that, for all $t_N \in S^*$ it is always possible to allocate all agents their targets. Moreover, these are the only profiles where this is true. Thus we have, $t_N \in S^* \iff \min_{\alpha\in\mathscr{A}^{IR}(t_N)} \max_{i\in N} \delta^{\alpha}(t_N) = 0.$

The performance of an assignment rule Ψ is then evaluated in the worst case, by computing the largest regret of Ψ overall target profiles such that the denominator in (1.13) is not 0.

Definition 1.4.2. *The worst case regret of an assignment rule* Ψ *is given by*

$$MR\big(\Psi\big) \;:=\; \max_{t_N \,\in\, S^{|N|}\setminus S^*} \; Regret\big(\Psi, t_N\big)$$

Lemma 1.4.1. *If an assignment rule* Ψ *is Pareto efficient then,*

$$MR\big(\Psi\big) \leq \frac{|N|-1}{1}$$

The proof appears in Appendix A.

Remark 1.4.2. *The worst faring efficient rule with respect to minimizing max delay is one that delivers an allocation with max delay of* $|N|-1$ *at some profile of targets, where it is in fact feasible to construct an allocation that has a max delay of 1.*

Proposition 1.4.5. *If an assignment rule* Ψ *is Pareto efficient, strategy-proof and non-bossy, then* $MR(\Psi) = |N|-1$, *i.e., it is the worst rule with respect to minimizing max delay.*

Proof. Fix a Pareto efficient, strategy-proof and non-bossy rule Ψ. Consider the profile $t_N = (t_i)_{i\in N}$ where $t_i = 1\ \ \forall i \in N$. Let $|N| = n$. Without loss of generality, suppose $\Psi_i(t_N) = i\ \ \forall i \in N$. Next, consider profile $t_N^2 = (t_2', t_{-2})$ which differs from t_N in that agent 2 now reports $t_2' = 2$. Strategy-proofness implies $\Psi_2(t_N^2) = 2 = \Psi_2(t_N)$. Then by non-bossiness $\Psi(t_N^2) = \Psi(t_N)$. Next, consider profile $t_N^3 = (t_2', t_3', t_{-\{2,3\}})$ which differs from t_N^2 in that agent 3 now reports $t_3' = 3$. Again by strategy-proofness and non-bossiness we have, $\Psi(t_N^3) = \Psi(t_N^2) = \Psi(t_N)$. Proceeding in the same manner by moving targets of agents, one at a time (target of agent i is moved from slot 1 to slot i every time), finally, we get $\Psi(t_N^{n-1}) = \Psi(t_N)$. Max delay of Ψ at this profile t_N^{n-1} is clearly $n-1$. However the allocation α such that $\alpha(1) = 1$, $\alpha(n) = 2$ and $\alpha(i) = i+1\ \ \forall i \in \{2,3,\cdots,n-1\}$ has a max delay of 1. ■

Proposition 1.4.5 in tandem with Proposition 1.4.2 and Proposition 1.4.4 respectively gives us the following corollaries.

Corollary 1.4.1. *If an assignment rule* Ψ *is Pareto efficient and group strategy-proof, then we have* $MR(\Psi) = |N|-1$.

Corollary 1.4.2. *The fixed priority rule is one of the worst rule with respect to minimizing max delay. For any fixed priority σ over agents, $MR(\psi_{\sigma}^{FP}) = |N| - 1$.*

The first come first served rule fares much better when it comes to max delay concerns (Proposition 1.4.6). However, it is not immune to manipulations of any kind (Proposition 1.4.7). This is not surprising given the tension between the two concerns which we have earlier highlighted in Proposition 1.4.3.

Proposition 1.4.6. *Every first come first served rule is Pareto efficient and moreover leximax minimizes delays.*

Proposition 1.4.6 is proved in Appendix A.

Proposition 1.4.7. *The first come first served rule fails to be*

(i) Strategy-proof

(ii) Pairwise reallocation proof

(iii) Envy-proof

Proposition 1.4.7 is proved in Appendix A.

The slot-based priority rule also has appealing incentive properties and is a little more adaptable to scenarios when the designer must respect exogenously given priority orders for every slot (see, Proposition 1.4.8 and Proposition 1.4.9 below). However, it suffers from the same drawback as the fixed priority rule when it comes to max delay concerns (Proposition 1.4.10).

Proposition 1.4.8. *Every slot-based priority rule is*

(i) Pareto efficient

(ii) Group strategy-proof

Proposition 1.4.8 is proved in Appendix A.

Proposition 1.4.8 along with Corollary 1.4.1 then gives us the following corollary.

Corollary 1.4.3. *The slot-based priority rule is one of the worst rules with respect to minimizing max delay, i.e., for any profile of priority orders* $\gg$, $MR(\psi_{\gg}^{SBP}) = |N| - 1$.

In applications, where the *profile of priority orders* $\gg = (\gg_s)_{s \in S}$ for slots is exogenously given rather than it being a design element, one can talk about the notion of *justified envy-freeness*, which is defined below.

Fix any profile of priority orders $\gg = (\gg_s)_{s \in S}$. An allocation $\alpha \in \mathscr{A}$ is said to **eliminate justified envy** at the profile of targets t_N given $\gg$ if:

- α is individually rational at t_N
- There do not exist agents i, j and slot s such that:

$$\alpha(j) = s, \quad t_i \leq s < \alpha(i), \quad \text{and} \quad i \gg_s j$$

Definition 1.4.3. *Fix any profile of priority orders* $\gg = (\gg_s)_{s \in S}$. *An assignment rule* Ψ *is* ***justified envy-free*** *at* $\gg$ *if, at every profile of targets* t_N*, the allocation* $\Psi(t_N)$ *eliminates justified envy at* t_N *given* $\gg$.

In words, an agent i is said to have justified envy towards agent j if the latter is allocated a slot for which the former has a higher priority. An assignment rule is justified envy-free if such a scenario never arises in its proposed allocation.

Remark 1.4.3. *Note that, when priority orders for slots are exogenously given, the appropriate definition of a mechanism or an assignment rule should then additionally include the space of priority orders in its domain. We have chosen to not do so here in order to keep things coherent, and therefore in the above definition we say an assignment rule is "justified envy-free at* $\gg$*" instead of "justified envy-free".*

It is a well known fact that the DA algorithm is stable Gale and Shapley (1962). This in turn implies, the agent proposing DA (taking the priority structure as preferences of the non-proposing side) is justified envy-free (see, Ergin 2002 for a discussion[5]). Invoking Remark 1.3.1 then gives us the following proposition.

Proposition 1.4.9. *Given any exogenous profile of priority orders* $\gg = (\gg_s)_{s \in S}$*, the slot-based priority rule* $\psi^{SBP}_{\gg}$ *is justified envy-free at* $\gg$.

Proposition 1.4.10. *The slot-based priority rule fails to be*

(i) Pairwise reallocation proof

(ii) Envy-proof

Proposition 1.4.10 is proved in Appendix A.

1.5 Concluding remarks

In this chapter, we have studied the problem of assigning agents to slots that are arranged on a line. Agents are born at different points on the line and want a slot as soon as possible, once they are born. Important considerations in this context like minimizing in various ways the delay faced by an agent, elicitation of truthful reports by participating agents, and efficiency of the overall system have been studied from an axiomatic standpoint. Our results indicate that the attractive incentive compatibility property of strategy-proofness is incompatible with individual delay concerns such as minimizing maximum delay or leximax minimizing the vector of delays. If individual delay concerns are of greater importance, we propose the implementation of the *first come first served rule*; while if the designer deems incentive compatibility as being more important, the *fixed priority rule* and the *slot-based priority rule* are the better choices. The slot based priority rule is a

[5] In their paper, they use the terminology: α adapts to $\gg$ to mean α is justified envy-free at $\gg$ given a profile t_N.

little more flexible when the designer must respect exogenous slot specific priorities over agents. The main results from our model readily extend to the scenario where slots have multiple (possibly different) capacities.

Future work can consider an interesting extension where agents don't just have different birth dates, but also have different deadlines after which they have no use of a slot; i.e., they are only interested in a particular segment of the line, and within that segment they want a slot as early as possible. Another fruitful direction might be to study probabilistic mechanisms on the same domain.

1.6 Appendix A - Proofs

Proof of Lemma 1.2.1

Let α be a Pareto efficient allocation at profile t_N. Next, assume for contradiction that there exists an unassigned slot $s \in S$ and an agent i with target $t_i \leq s$ such that $\alpha(i) > s$. Then, the new allocation α' such that $\alpha'(i) = s$ and $\alpha'(j) = \alpha(j)\ \forall j \neq i$ Pareto dominates the allocation α, a contradiction.

To show the other direction, let α be an allocation at profile t_N and suppose there does not exist an unassigned slot $s \in S$ and an agent i with target $t_i \leq s$ such that $\alpha(i) > s$. Assume for contradiction that α is not Pareto efficient. Suppose allocation α' Pareto dominates α. Let i_1 be that agent who is made strictly better and suppose $\alpha'(i_1) = s^{i_1}$. This implies, $t_{i_1} \leq s^{i_1} < \alpha(i_1)$. Moreover, this slot s^{i_1} was originally assigned to some agent at allocation α by assumption. Suppose that agent is i_2, i.e., $\alpha(i_2) = s^{i_1}$. This means i_2 must be made weakly better under α' as well. But since slot s^{i_1} is now allocated to i_1 at α', agent i_2 must get an even better slot in α'. Thus we have, $t_{i_2} \leq s^{i_2} < \alpha(i_2)$. Combining the above relations we get, $t_{i_2} \leq s^{i_2} < \alpha(i_2) < \alpha(i_1)$. Proceeding in this manner leads to one of the two following cases: (1) an agent who is allocated her target slot in α must be allocated a better slot, which is impossible by definition; or, (2) we reach the first slot in the line in which case the scenario in (1) is necessarily true. Both these cases, therefore, leads to a contradiction that α' Pareto dominates α. ■

Proof of Lemma 1.2.3

Fix a profile of targets $t_N = (t_i)_{i \in N}$. Let $\alpha \in \mathscr{A}$ be a Pareto efficient allocation at t. There is a partition of S satisfying the conditions stated in Lemma 1.2.2. Let it be $\{S_0, S_1, S_2, \cdots\}$. Next consider all slots that are assigned, i.e., slots in S_k such that $k \in \{1, 3, 5, \cdots\}$. The target slot of every agent at profile t_N belongs to one such S_k, i.e., for all i, $t_i \in S_k$ for some $k \in \{1, 3, 5, \cdots\}$. Now, consider any such S_k. Let $\alpha^{-1}(S_k)$ be the set of individuals who are assigned to a slot in S_k under α. Since α is Pareto efficient, this implies that for every agent $i \in \alpha^{-1}(S_k)$ her target $t_i \in S_k$ (By Lemma 1.2.2, $\alpha_i \in S_k \iff t_i \in S_k$). Now, since every slot in S_k is assigned, we have precisely $|S_k|$ such agents: $|\alpha^{-1}(S_k)| = |S_k|$. Next, suppose $\tilde{\alpha}$ is an allocation that minimizes aggregate delay. Since S_k contains consecutive slots, and all slots in S_k are needed to accommodate agents in $\alpha^{-1}(S_k)$, allocation $\tilde{\alpha}$ must therefore also have assigned agents in $\alpha^{-1}(S_k)$ to some slot in S_k (these are the closest slots to $\alpha^{-1}(S_k)$ collectively speaking). As a result, allocation α can be obtained from $\tilde{\alpha}$ by performing a sequence of pairwise swaps between agents belonging to the same set $\alpha^{-1}(S_k)$. Each such pairwise swap reduces the delay of one agent in the pair at the expense of the other agent. As a result, there is no change in aggregate delay after each such swap. Since $\tilde{\alpha}$ minimizes aggregate delay, therefore so must α.

Next let α' be an allocation that minimizes aggregate delay at profile of targets t_N. Assume for contradiction that it is not Pareto efficient. This implies that there is an allocation α'' such that $0 \leq \delta_i(\alpha'', t_i) \leq \delta_i(\alpha', t_i)$ for every agent i, with at least one strict (second) inequality. This implies $\sum_{i \in N} \delta_i(\alpha'', t_i) < \sum_{i \in N} \delta_i(\alpha', t_i)$, the desired contradiction. ■

Proof of Lemma 1.2.5

Fix a profile $t_N = (t_i)_{i \in N}$. Let $\alpha \in \mathscr{A}$ be a Pareto efficient allocation at t_N. There is a partition of S satisfying the conditions stated in Lemma 1.2.2. Let it be $\{S_0, S_1, S_2, \cdots\}$. Consider any set S_k such that $k \in \{1, 3, 5, \cdots\}$. Let $\alpha^{-1}(S_k)$ be the set of individuals who are assigned to a slot in S_k under α. Since α is Pareto efficient, this implies that for every agent $i \in \alpha^{-1}(S_k)$ her target $t_i \in S_k$ (By Lemma 1.2.2, $\alpha_i \in S_k \iff t_i \in S_k$). Now, since every slot in S_k is assigned, we have precisely $|S_k|$ such agents: $|\alpha^{-1}(S_k)| = |S_k|$. Next, suppose $\tilde{\alpha}$ is an allocation that leximax minimizes delay. Since S_k contains consecutive slots, and all slots in S_k are needed to accommodate agents in $\alpha^{-1}(S_k)$, allocation $\tilde{\alpha}$ must therefore also have assigned agents in $\alpha^{-1}(S_k)$ to some slot in S_k (these are the closest slots to $\alpha^{-1}(S_k)$ collectively speaking). As a result, allocation $\tilde{\alpha}$ can be obtained from α by performing a sequence of pairwise swaps between agents belonging to the same set $\alpha^{-1}(S_k)$. Since, assigned slots are ordered like targets in α, when performing the pairwise swaps just described, there is no change in the vector of delays corresponding to the two allocations α and α'. This implies that α also leximax minimizes delays. ■

Proof of Proposition 1.4.4

Fix a priority σ, and consider ψ_σ^{FP} - the associated fixed priority rule. Recall, the *best slot* among a set of slots $Y \subseteq S$ given a target slot t_i is denoted as

$$B\big(t_i,\, Y\big) \;=\; \min\,\{\, s' \in Y \mid s' \geq t_i \,\}$$

Next we show that ψ_σ^{FP} is strategy-proof. The strategy of any agent i is the target t_i that she reports. Suppose agent i wants to deviate. When agent i is truthful, let S^{-i} be the set of slots allocated to agents who have higher priority than i $\big($an agent j has higher priority than agent i if and only if $\sigma^{-1}(j) < \sigma^{-1}(i)\big)$. So, by being truthful, agent i gets $B(t_i,\, S \setminus S^{-i})$. When agent i deviates, any agent j who has a higher priority than agent i continues to get the same slot that he was getting when agent i was truthful. So, agent i still gets a slot in $S \setminus S^{-i}$. Hence, the deviation cannot be better.

To see that ψ_σ^{FP} is non-bossy, fix a profile of targets $t_N = (t_i, t_{-i})$ and an agent i. Again, let S^{-i} be the set of slots allocated to agents who have higher priority than i. Then allocation of agent i at profile $t_N = (t_i, t_{-i})$ is $\Psi_i(t_N) = B(t_i,\, S \setminus S^{-i})$. Note that, agent i by deviating to t_i' can never change the allocation of agents who gets a slot in S^{-i}. Now, if by deviating to t_i', agent i gets the same slot, i.e., $\Psi_i(t_i', t_{-i}) = \Psi_i(t_N)$, then the set of slots available to agents who has a lower priority than agent i is invariant in both profiles t_N and (t_i', t_{-i}). Since the reports of other agents are fixed, we have that if agent i's allocation does not change there is no change in the allocations of other agents.

Since, ψ_σ^{FP} is both strategy-proof and non-bossy, by Proposition 1.4.2, ψ_σ^{FP} is group strategy-proof.

To show Pareto efficiency, assume for contradiction that ψ_{σ}^{FP} is not Pareto efficient. Consider a profile t_N such that $\psi_{\sigma}^{FP}(t_N) = \alpha$. Then, there is another allocation α' satisfying $0 \leq \delta_i(\alpha', t_i) \leq \delta_i(\alpha, t_i)$ for every agent i, with at least one strict inequality (for the second inequality). Now, consider the first agent j in the priority σ such that $0 \leq \delta_j(\alpha', t_j) < \delta_j(\alpha, t_j)$. This implies, $t_j \leq \alpha'(j) < \alpha(j)$. Since agents before j in the priority σ have the same delay in both α and α' (i.e., they are assigned the same slot in both α and α'), the slot $\alpha'(j)$ was still available to be assigned to agent j. This is a contradiction since $\alpha(j)$ was the best available slot for agent j but $t_j \leq \alpha'(j) < \alpha(j)$.

Assume for contradiction that ψ_{σ}^{FP} is not pairwise reallocation proof. Then, ψ_{σ}^{FP} is strongly manipulable through pairwise reallocation. This implies, there exists a profile of targets $t_N = (t_i, t_j, t_{-\{i,j\}}) \in S^{|N|}$, distinct agents $i, j \in N$, and $t'_i, t'_j \in S$ such that

(i) $t_i \leq \Psi_j(t'_i, t'_j, t_{-\{i,j\}}) < \Psi_i(t_N)$

(ii) $t_j \leq \Psi_i(t'_i, t'_j, t_{-\{i,j\}}) < \Psi_j(t_N)$

Next, without loss of generality, let us assume that $\sigma^{-1}(i) < \sigma^{-1}(j)$. When agent i is truthful (i.e., she reports t_i), let S^{-i} be the set of slots allocated to agents who have higher priority than agent i. Now, since $\sigma^{-1}(i) < \sigma^{-1}(j)$ and we have the other agents' reports held fixed, agents i and j therefore can only get slots from the set $S \setminus S^{-i}$. That is to say, for all $t'_i, t'_j \in S$ it must be the case that $\Psi_j(t'_i, t'_j, t_{-\{i,j\}}) \in S \setminus S^{-i}$. Now, by being truthful, agent i gets $\Psi_i(t_N) = B(t_i, S \setminus S^{-i})$, which is the best available slot for agent i in $S \setminus S^{-i}$. But, from condition (i), we have $t_i \leq \Psi_j(t'_i, t'_j, t_{-\{i,j\}}) < \Psi_i(t_N)$, which is a contradiction.

Next show that ψ_{σ}^{FP} is envy-proof. Fix a profile of targets $t_N = (t_i)_{i \in N}$ and an agent i. When agent i reports t_i, S^{-i} is the set of slots allocated to agents who have higher priority than agent i. Then, agent i at profile t_N gets $\Psi_i(t_N) = B(t_i, S \setminus S^{-i})$, the best available slot. This means agent i does not envy any agent who gets a slot in $S \setminus S^{-i}$. Equivalently stating, if agent i envies agent j, then agent j was allocated a slot in S^{-i}, i.e., $\Psi_j(t_N) \in S^{-i}$. But, $\Psi_j(t_N) \in S^{-i}$ implies $\sigma^{-1}(j) < \sigma^{-1}(i)$. This in turn implies $\Psi_j(t_N) = \Psi_j(t_i', t_{-i})$ for all $t_i' \in S$, since the allocation of an agent higher in the priority order is not affected by the reports of agents lower down the order. Thus, agent i can never harm an agent she envies. Since we can repeat the same argument for any profile of targets and any agent, we conclude that Ψ is envy-proof. ■

Proof of Lemma 1.4.1

Recall that $MR(\Psi) = \max_{t_N \in S^{|N|} \setminus S^*} Regret(\Psi, t_N)$

where, $Regret(\Psi, t_N) = \dfrac{\max_{i \in N} \delta^{\Psi(t_N)}(t_N)}{\min_{\alpha \in \mathscr{A}^{IR}} \max_{i \in N} \delta^{\alpha}(t_N)}$.

Since the assignment rule Ψ is Pareto efficient, by Lemma 1.2.1 there must not be any used slots in the allocation output by Ψ. The greatest value of the numerator in $Regret(\Psi, \cdot)$ occurs at the profiles where all agents have the same target slot. This is because, at these profiles, only one agent can be made fully happy, leading to a max delay of $|N| - 1$ necessarily, if there are $|N|$ agents. In other profiles, more than one agent can be given their best slot thereby reducing max delay to be strictly less than $|N| - 1$, as there is an arbitrary number of slots. Furthermore, the minimum value for the denominator of the regret expression is 1 since only profiles $t_N \notin S^*$ are considered. ■

Proof of Proposition 1.4.6

Fix a priority σ, and consider ψ_σ^{FCFS} - the associated first come first served rule. Next consider a profile t_N and the associated first come first priority σ_{FF}. Now, $\psi_\sigma^{FCFS}(t_N) = \psi_{\sigma_{FF}}^{FP}(t_N) = \alpha$ (say). By Proposition 1.4.4 we have that α is Pareto efficient. Next, recall that $t_i < t_j \implies \sigma_{FF}^{-1}(i) < \sigma_{FF}^{-1}(j)$. Also, for every $Y \subseteq S$ we have $t_i < t_j \implies B(t_i, Y) \leq B(t_j, Y)$[6]. Together this then implies, $\alpha_i < \alpha_j$. Finally, applying Lemma 1.2.5 the result follows. ■

[6] $B(t_i, Y) = \min\{s' \in Y \mid s' \geq t_i\}$, denotes the *best slot* among a set of slots $Y \subseteq S$ given a target slot t_i.

Proof of Proposition 1.4.7

Consider a problem with three agents $N = \{1,2,3\}$ and three slots $S = \{a,b,c\}$ where a and c are the first and last slot respectively. A profile $t_N = (t_1,t_2,t_3)$ is denoted simply as $t_1t_2t_3$ and an allocation $\Psi(t_N)$ at profile t_N where $\Psi_1(t_N) = a$, $\Psi_2(t_N) = b$ and $\Psi_3(t_N) = c$ is denoted as $\Psi(t_N) = abc$.

Let us fix an order σ such that: $\sigma(i) = i$ for all i. Now, consider ψ_σ^{FCFS}- the associated first come first served rule.

To see that ψ_σ^{FCFS} is not strategy-proof, consider the profile $t_N = aba$. The rule at this profile outputs $\psi_\sigma^{FCFS}(t_N) = acb$, where agent 2 has a delay of 1. At profile $t'_N = aaa$, we have $\psi_\sigma^{FCFS}(t'_N) = abc$. Therefore, agent 2 deviates at t_N by misreporting $t'_2 = a$ to get a delay of 0.

To see ψ_σ^{FCFS} is not envy-proof, consider profile the profile of targets $t''_N = baa$. We have $\psi_\sigma^{FCFS}(t''_N) = cab$ and thus agent 1 envies agent 3's slot. By misreporting $t'_1 = a$, agent 1 can hurt agent 3, since $\psi_\sigma^{FCFS}(t'_N) = abc$. Thus, ψ_σ^{FCFS} is not envy-proof.

Next, suppose there are two more agents and slots: $N = \{1,2,3,4,5\}$ and $S = \{a,b,c,d,e\}$. Priority order σ is such that: $\sigma(i) = i$ for all i. To see that ψ_σ^{FCFS} is not pairwise reallocation proof, consider profile $t_N = abbbb$. We have, $\psi_\sigma^{FCFS}(t_N) = abcde$. Now, if both agents 4 and 5 deviate to $t'_4 = t'_5 = a$, they get slot b and c respectively, which they already prefer to their slots at profile t_N. But even if they swap these new allocations ex-post, they both still strictly prefer their new allocations to their initial allocation by being truthful. Hence, ψ_σ^{FCFS} is not pairwise reallocation proof. ■

Proof of Proposition 1.4.8

Fix a profile of priority orders $(\gg_s)_{s\in S}$ and a profile of targets $t_N = (t_i)_{i\in N}$. Let α denote the allocation produced by the slot-based priority rule.

Now, any slot $k \in \{1,2,3,\ldots\}$ is unassigned at this profile only if, in **Step k** of the algorithm we have $I^k \setminus \bigcup_{j=1}^{k-1} i^j = \emptyset$ [7] where, $I^k = \{i \in N \mid t_i \leq k\}$ and $i^1, i^2, \cdots, i^{k-1}$ are (if $i^j \neq \emptyset$) agents with targets strictly less than k who are already assigned an earlier slot. Therefore, if slot k is unassigned then for every agent i with target $t_i \leq k$ we have that $\alpha_i \leq k$. Then, from Lemma 1.2.1 we have that α is Pareto efficient.

Since every slot-based priority rule is Pareto efficient, using Remark 1.3.1 and invoking Theorem 1 in Ergin (2002) it is also group strategy-proof. ■

Proof of Proposition 1.4.10

Consider a problem with three agents $N = \{1,2,3\}$ and three slots S ={a, b, c} where a and c are the first and last slot respectively.

Suppose slots a and b have the following priorities:

$$1 \gg_a 3 \gg_a 2$$

$$2 \gg_b 1 \gg_b 3$$

Then, at profile $t_N = aba$, we have the allocation $\psi^{SBP}_{\gg}(t_N) = abc$.

At profile $t'_N = baa$, we have $\psi^{SBP}_{\gg}(t'_N) = cba$.

Agents 1 and 2 jointly misreport at profile $t'_N = baa$ to profile $t_N = aba$ and swap allocations ex-post, which they both strictly prefer to what they get by being truthful.

To see that the rule is not envy-proof, suppose next that slots a and b have the following priorities:

$$1 \gg_a 2 \gg_a 3$$

$$2 \gg_b 3 \gg_b 1$$

Now at profile $t''_N = aaa$ we have $\psi^{SBP}_{\gg}(t''_N) = abc$. At profile $t'_N = baa$ we have $\psi^{SBP}_{\gg}(t'_N) = cab$ where agent 1 envies agent 3. Thus, by misreporting to $t''_1 = a$, agent 1 can hurt agent 3. ■

[7] $i^0 = \emptyset$

Chapter 2

Stable matchings with indifferences and the priority rule

Abstract

In this chapter, we study a two-sided one-to-one market setting where one side of the market is objectively ranked. Constraints prevent agents on the other side from matching with agents who are ranked higher than an agent-specific threshold. The preference domain allows for indifferences, with preferences of one side being derived from a master preference list, while agents on the other side have general preferences. We analyze matchings with respect to two standard properties: *stability* and *Pareto efficiency*. Having met these standard goals, the designer facing this problem may additionally be interested in a third criterion: the *size* of the matching. We show that for the proposed domain all stable matchings have the same size. We propose an assignment rule that always produces a stable and Pareto efficient matching. Furthermore, we show that every matching that is both stable and Pareto efficient is hierarchical by nature in a precise sense. The domain has a variety of practical applications including the refugee matching market, which has recently received a lot of attention in the literature.

2.1 Introduction

Let us consider a two-sided one to one market setting, where agents on one side of the market (say "Projects") are grouped into indifference classes that are exogenously ranked, and agents on the other side (say "Firms") prefer higher-ranked projects. However, each firm face internal feasibility constraints that prevent her from matching with projects ranked above an individual threshold. Examples include — (1) the refugee matching problem: where *benevolent hosts* (firms) with a limited beds to spare, are willing to shelter *refugee families* (projects) of various sizes in their homes, in exchange for a monetary reward (which is increasing in family size) from the government; (2) firms competing for governmental projects over which there is generally a consensus as to which ones are more desirable, but firm-specific capacity constraints (technological, deadline, budget etc.) prevent it from going after very top projects; (3) the slot allocation problem: where *slots* (projects) are arranged on a line with earlier slots ranked higher, *users* (firms) arrive over time and wish to be served as early as possible once they do, for instance, researchers wishing to use a supercomputer, ships wanting to load/unload cargo at a port, etc. Being indifferent between different options is a ubiquitous phenomenon in our society, and the above examples are no exception.

These observations collectively inform our assumptions on the preference domain. We assume that projects are exogenously ranked (but several projects can have the same rank, thus giving rise to indifference classes), and each firm f faces a threshold t_f. This firm f prefers projects of higher rank, but projects of rank above t_f are not acceptable to her. Preferences of firms thereby have a nested structure (formally presented in Section 2.2.1). Each project j can have arbitrary (non-strict) preferences over firms but prefers any firm over remaining unmatched. Taking preferences of both sides into account, firms must be matched to projects in a one-to-one fashion.

We adopt a mechanism design approach to solve the problem. We will pursue two important design goals. The first is *stability*, which ensures that agents would accept the matching proposed by the designer and would not try to individually renegotiate it. To define it, we rely on the notion of *strong blocking*: two agents form a pair and block a matching if each agent in the pair strictly prefers the other over their current match[1]. The second is *Pareto efficiency*, i.e., the matching cannot be Pareto improved upon at any preference profile. Having met these two goals, in the above examples, the designer may additionally wish to improve welfare by maximizing as much as possible the *size* (i.e., the total number of matched pairs) of the proposed matching.

In our domain with indifferences, all stable matchings have the same size (Proposition 2.2.1). To the best of our knowledge, it is the first non-trivial non-strict preference domain in the literature for which this fact holds. Thus, in the space of stable and Pareto efficient matchings, the designer is not constrained by welfare considerations regarding the matching size. This is not just a nice to have result but is rather quite significant due to its practical implications. This is because, in the presence of indifferences (or, ties), finding a stable matching that maximizes size is a computationally hard problem even under restrictions on the number and length of ties (Manlove et al., 2002), or preferences of both sides being limited to a master list (Irving et al., 2008). For our proposed domain that admits a wide range of applications, the designer does not face this challenge since there are efficient algorithms for finding a stable matching.

Our mechanism design problem involves matching agents to each other without the possibility of resorting to money or lotteries (or, fractional assignments). As a consequence, most fairness requirements are immediately out of reach. Any matching will therefore favour some agents and leave others largely unsatisfied. In such settings, it is widely observed that resulting mechanisms tend to have a hierarchical structure. Formally proving, or even formalizing this statement is however elusive. Under strict preferences some results are obtained: see, for instance, pure assignment models without transfers and exchange models in Svensson (1999), Pápai (2000) and Pycia and Ünver (2017). Under non-strict preferences, we are only aware of Svensson (1994) and Bogomolnaia et al. (2005). Most of these pa-

[1]This notion is called *weak stability* in the literature, but we refer to it simply as *stability*. Irving (1994) proposes two other natural definitions of stability in the presence of indifferences: *strong stability* and *super stability*. These are stronger notions, but contrary to weak stability, they are not guaranteed to exist.

pers attempt to characterize the set of incentive compatible (in a strong sense of strategy-proofness) and efficient mechanisms and demonstrate that they have a hierarchical structure. In our more general setup, we are not able to obtain such a clean characterization result, however, we show that stable and efficient matchings are still arguably hierarchical by nature.

Hierarchical nature of stable and efficient matchings

We propose an assignment rule (mechanism), which we call the *decreasing refined priority rule* (Definition 2.3.5), that always generates a stable and Pareto efficient matching for any given problem (Proposition 2.3.2). The rule is in the spirit of the classical serial dictatorship rule where an ordering of projects (in the queue) is used to determine the allocation of firms: the first project in the queue picks his best firm and leaves with that firm, from what remains the second project in the queue picks, from what remains the third project in the queue picks and so on. Thus, the serial dictatorship rule starts with the set of all firms, and at every step eliminates a firm based on preferences of projects. The *decreasing refined priority rule* adds two additional tweaks to this procedure:

* Firstly, the priority (queue) order over projects used by the rule is always aligned with the exogenous ranking over the indifference classes of projects: the projects in the top indifference class appear first in the queue order, followed next by projects in the second-from-top indifference class, and so on (see, Definition 2.3.2). This ensures stability.

* Secondly, the rule starts with the set of matchings instead of the set of firms. Following the queue order, at every step, the rule eliminates all[2] matchings where the project being assigned does not get his best firm (one of his best firms if there are many), from the set of matchings available at that step. This refinement ensures that the resulting matching is Pareto efficient, given that preferences exhibit indifferences. This point is elaborated further in Example 2.3.1 and the discussion thereafter.

[2] As opposed to eliminating only one firm from the set of available firms.

Stability and Pareto efficiency do not however characterize the decreasing refined priority rule (Proposition 2.3.3). Nevertheless, we show that every matching that is both stable and Pareto efficient can be generated by a decreasing (non-refined) priority rule. Therefore, in our proposed domain, stable and Pareto efficient matchings are hierarchical by nature: they can be thought of as an outcome of projects being arranged in a queue order, with each project receiving one of their best firms in turn, from the set of remaining firms (Proposition 2.3.5).

In Section 2.4, we discuss a strengthening of the stability notion to *strong stability*. We show that strongly stable matchings (those matchings where there are not even *weak blocking* pairs, where only one member of the pair strictly benefits) exist very rarely, and imply a very specific structure of both preferences and the strongly-stable matchings themselves. If they do exist at some preference profile, then the set of all strongly stable matchings at that profile is exactly equal to the set of all matchings that can be generated by a decreasing refined priority rule at that profile (Proposition 2.4.1).

2.1.1 Relation to the existing literature

It is well known that, in a bilateral matching problem, when every agent has a strict (but arbitrary) preference over agents on the other side, the *size* of all stable matchings are *identical*, a consequence of the *Lone Wolf Theorem*[3] (McVitie and Wilson, 1970; Gale and Sotomayor, 1985). This result breaks down if we allow for general preferences (Roth and Sotomayor, 1990). However, in our domain with indifferences, the equivalence of stable matchings with respect to *size* remains true (Proposition 2.2.1).

This paper contributes to the literature on matching with indifferences. In particular, it is closely related to the Stable Marriage problem with Ties and Incomplete Lists with a Master List (Irving et al., 2008). The term "Ties" simply means indifferences in our context; "Incomplete Lists" refer to the fact that each agent's

[3]The theorem illustrates that the set of unmatched agents does not depend on the choice of stable matching.

preference list may consist of only a subset of the members of the other side (the *acceptable* partners of this agent); and an agent's preference list contains her acceptable partners ranked precisely according to the "Master List". In the above problem, Irving et al. notes that even when preferences of both sides are derived from a master list[4], weakly stable matchings need not have the same size, and subsequently discusses the algorithmics of finding weakly stable matchings that maximizes size. In our model, where preferences of only one side are derived from a master list, while preferences of the other side are not incomplete[5]; weakly stable matchings necessarily have the same size (Proposition 2.2.1). This result is of practical significance since, in the presence of incomplete lists and ties (indifferences), finding a weakly stable matching that maximizes size is a hard problem; even under restrictions on the number and length of ties (Manlove et al., 2002), or preferences of both sides being limited to a master list (Irving et al., 2008). For a comprehensive survey on algorithmic results related to matching with indifferences we refer the reader to Manlove (2013).

On structural results, Manlove (2002) proves the lattice structure of the set of strongly stable matchings with ties (without allowing for incomplete lists) and notes the absence of such a structure for the set of weakly stable matchings. Our structural results, therefore, do not concern the set of weakly stable matchings. We argue instead that, every matching that is both (weakly) stable and efficient admits a hierarchical structure in a precise sense (Proposition 2.3.5).

Other matching applications with indifferences include: the school choice problem (Erdil and Ergin 2008, 2017); the housing market where agents are initially endowed with a house (Aziz and De Keijzer, 2012); kidney exchange models (Roth et al., 2005; Andersson and Kratz, 2020).

Our setting can be considered as an extension of the "house allocation" model. Here the role of "houses" is played by "firms", but contrary to that model, firms

[4]Note that, there are two master lists, one for each side of the market.

[5]We assume that each *project* prefers any *firm* over remaining unmatched.

now have preferences, albeit rather homogeneous ones (see, Section 2.2.1). Our model is reduced to the "house allocation" model with arbitrary (non-strict) preferences when all projects have the same rank. In this reduced model, Bogomolnaia et al. (2005) showed that the set of all stable and efficient assignments is equal to the set of all assignments obtained by some *refined priority rule*[6]. In our more general case, we do not obtain such a crisp equivalence result, though our results have a similar flavour. A *decreasing refined priority rule* always outputs a stable and efficient matching (Proposition 2.3.2). Every matching that is both stable and efficient is a result of a *decreasing (non-refined) priority rule* (Proposition 2.3.5). If a preference profile admits a strongly stable matching[7], then the set of all strongly stable matchings is equal to the set of all matchings that can be generated by some decreasing refined priority rule (Proposition 2.4.1).

Finally, the preference domain studied in this paper is also related to that of the Refugee Matching problem discussed by Andersson and Ehlers (2019). In their model, *refugee families* (projects) are also ordered according to family size giving rise to blocks containing projects of the same size. *Hosts* (firms) have limited beds and therefore their preferences have a nested structure[8] over these blocks similar to ours. The differences are as follows.

Firstly, in our domain, hosts are necessarily indifferent between all refugee families in the same block (as a result we have indifference classes); while in their model, hosts can have general preferences over refugee families within the same block while respecting the nested structure between blocks.

Secondly, contrary to our domain, their model does not elicit any preferences from refugee families whatsoever. It is important to take into account their preferences: there is growing evidence that the initial placement of refugee families greatly affects outcomes like education, job prospects, and earnings — which in turn profoundly alters their lifetime welfare, as most refugees do not move from

[6]Since all projects have the same rank, the decreasing priority order plays no role.
[7]The existence of a strongly stable matching is however a rare event. For a detailed discussion, see Section 2.4.
[8]Hosts prefer refugee families of greater size subject to the condition that the host has sufficient number of beds.

the localities to which they are resettled for many years (Åslund and Rooth (2007), Damm (2014), Jones and Teytelboym (2017), Åslund et al. (2010), Martén et al. (2019)). As described in Jones and Teytelboym (2018), ignoring families' individual preferences has even caused families seeking shelter in Finland to cancel their asylum applications.

2.1.2 Outline of the chapter

In Section 2.2 we present the formal model and the result that stable matchings have the same size. Section 2.3 presents the formal definitions for candidate assignment rules and evaluates them with respect to stability and efficiency. Section 2.4 discusses the strengthening of stability to strong stability. A concluding discussion follows. Some proofs are relegated to the appendix.

2.2 Model

Throughout the rest of this chapter, we use the project-firm terminology for convenience. We take the liberty to personify projects and firms, and refer to them using pronouns "he" and "she" respectively. There is a finite set $J = \{1, 2, ..., |J|\}$ of projects indexed by j and a finite set F of firms indexed by f. Furthermore, J is partitioned into a set of indifference classes $\{J_1, J_2, ..., J_p\}$, and there is an exogenously given (strict) order $\succ$ over these classes: say it is $J_p \succ J_{p-1} \succ ... \succ J_1$, without loss of generality. All the above ingredients are fixed for the rest of the chapter.

2.2.1 The Preference Structure

The preference relation of every firm $f \in F$ is completely identified by a threshold $t_f \in \{1, 2, ..., p\}$ that she reports. Let R^*_k be a weak order (complete and transitive relation) over $J \cup \{f\}$ $\left(f \in F\right)$ defined as below:

$$J_k \; P^*_k \; J_{k-1} \; P^*_k ... \; J_1 \; P^*_k \; \mathbf{f} \;\; P^*_k \; J_p \;\; I^*_k \; J_{p-1} \;\; ... \;\; I^*_k \; J_{k+1}$$

where, P_k^* and I_k^* denotes the strict and indifference part of R_k^* respectively and $k \in \{1,2,...,p\}$. Then, the preference relation of a firm $f \in F$ with threshold t_f, denoted by R_f is given by $R_{t_f}^*$, i.e., $R_f \equiv R_{t_f}^*$ where $t_f \in \{1,2,...,p\}$ and the **f** in the definition of $R_{t_f}^*$ is to be interpreted simply as the corresponding firm f. The preferences of firms have a nested structure in the sense that if for two firms f and f' we have $R_f \equiv R_k^*$ and $R_{f'} \equiv R_{k'}^*$ with $k > k'$, then over the indifference classes $J_{k'}, J_{k'-1}, ..., J_1$, both R_f and $R_{f'}$ are identical.

Preference of a project $j \in J$, denoted by R_j, is a weak order over the set $F \cup \{j\}$. Let P_j and I_j denote the antisymmetric and symmetric parts of R_j, respectively.

Let $R = (R_i)_{i \in J \cup F}$ denote a profile of preferences. Given a profile R, we will say that a firm f $\big($resp., project $j\big)$ is *acceptable* to a project j $\big($resp., firm $f\big)$ if fP_jj $\big($resp., $jP_ff\big)$. We will say that a project j and a firm f are *compatible* if they find each other mutually acceptable. We assume throughout that all projects find every firm acceptable, i.e., for any project $j \in J$, we assume that $fP_jj \;\; \forall f \in F$.

Finally, let $\mathscr{R}$ denote the set of all such preference profiles.

2.2.2 Notations and definitions

A **matching** is a function $\mu : J \cup F \to J \cup F$ such that:

- $\mu(j) \in F \cup \{j\} \; \forall \; j \in J$
- $\mu(f) \in J \cup \{f\} \; \forall \; f \in F$
- $\mu(f) = j \iff \mu(j) = f, \forall \; j \in J, f \in F$

Let $\mathscr{M}$ be the set of all possible matchings.

A matching $\mu \in \mathscr{M}$ is **individually rational** at profile $R \in \mathscr{R}$ if every agent is *acceptable* to his or her mate, i.e., $\nexists \, i \in J \cup F$ such that $iP_i\mu(i)$.

A matching $\mu \in \mathscr{M}$ is *strongly blocked* at profile $R \in \mathscr{R}$ by a pair $(j,f) \in J \times F$, if $jP_f\mu(f)$ and $fP_j\mu(j)$. A matching $\mu \in \mathscr{M}$ is *weakly blocked* at profile $R \in \mathscr{R}$ by a pair $(j,f) \in J \times F$, if $jR_f\mu(f)$ and $fR_j\mu(j)$ (with at least one of the relations being strict). A matching $\mu \in \mathscr{M}$ is **stable** at profile $R \in \mathscr{R}$, if it is individually rational and is not strongly blocked by any pair of agents. A matching $\mu \in \mathscr{M}$

is **strongly stable** at profile $R \in \mathscr{R}$, if it is individually rational and is not even weakly blocked by any pair of agents.

A matching $\mu \in \mathscr{M}$ is **Pareto efficient** (henceforth, **efficient**) at profile $R \in \mathscr{R}$, if there is no other matching μ', such that $\mu'(i) R_i \mu(i)\ \forall\ i \in J \cup F$ and $\mu'(i) P_i \mu(i)$ for at least one agent $i \in J \cup F$.

The **size** of a matching $\mu \in \mathscr{M}$ is defined as the number of matched pairs, i.e., the number of pairs $(j, f) \in J \times F$ such that $\mu(j) = f$.

Let us recall that J is partitioned into a set of indifference classes. Suppose we consider one such class J_k, $k \in \{1, 2, ..., p\}$. Now consider a project $j \in J_k$ and a firm f with $R_f \equiv R_s^*$ where $s < k$. Then, given the way the preference R_s^* is defined, project j is unacceptable to her. Then, although project j considers firm f as an acceptable match, *individual rationality* dictates that they are not matched to each other. Therefore, it will be useful to keep track of all firms that consider the class J_k as acceptable.

Given a preference profile R and some $k \in \{1, 2, ..., p\}$, let $C(J_k, R)$ denote the set of firms that find projects in J_k as *acceptable*, i.e., for every firm $f \in C(J_k, R)$ we have $R_f \equiv R_s^*$ where $s \geq k$ (therefore, $J_k\ P_f\ f$). Given the nested structure of preferences, for $k > k'$, every firm that considers J_k acceptable also finds $J_{k'}$ acceptable but not necessarily the other way round. As a result, we have an inclusion $C(J_k, R) \subset C(J_{k'}, R)$ for $k > k'$. In particular, $C(J_p, R) \subset C(J_{p-1}, R) ... \subset C(J_1, R)$. Let $J_{<k}$ be a shorthand to denote J_x with $x < k$. $J_{\leq k}$, $J_{\geq k}$ and $J_{>k}$ are similarly defined. Finally note that, every firm $f \in C(J_k, R)$ strictly prefers J_k over $J_{<k}$, i.e., we have $J_k\ P_f\ J_{<k}$ for every $f \in C(J_k, R)$.

Given a matching μ, for every indifference class J_k, $k \in \{1, 2, ..., p\}$, the set of projects in J_k who are matched under μ are collected in the set below:

$$J_k^{\mu} = \{j \in J_k \mid \mu(j) \neq j\}$$

The next proposition brings out the relationship between the size of all stable matchings at any given preference profile.

Proposition 2.2.1. *Given any profile of preferences $R \in \mathscr{R}$, any two* stable *matchings μ and μ' must have the same* size. *In particular, for every $k \in \{1,2,...,p\}$,*

$$|J_k^{\mu}| = |J_k^{\mu'}|$$

Proof. Consider first the top indifference class of projects J_p. Let μ and μ' denote two stable matchings such that $|J_p^{\mu}| > |J_p^{\mu'}|$, i.e., fewer projects from J_p are matched under μ' compared to μ. Since a project in J_p can only be matched to some firm in $C(J_p,R)$, this means that there are at least $|J_p^{\mu}|$ firms in $C(J_p,R)$. But only $|J_p^{\mu'}|$ of them are matched under μ'. This implies that there exists at least one firm $f \in C(J_p,R)$ who was matched to a project in J_p under μ but is either unmatched or is matched to a project in $J_{<p}$ under μ'. But $|J_p^{\mu}| > |J_p^{\mu'}|$ implies there is a project $j \in J_p$ who is matched under μ but not under μ'. Pair (j,f) would then block μ', a contradiction to the stability of μ'.

Next, let $k < p$ be the largest integer such that there are two stable matchings μ and μ' such that $|J_k^{\mu}| > |J_k^{\mu'}|$, i.e. fewer projects from J_k are matched in μ' than in μ. Therefore, both μ and μ' matches the same number of projects from classes, $J_{k+1}, J_{k+2}, ..., J_p$, i.e. we have $|J_q^{\mu}| = |J_q^{\mu'}| = X_q \ \forall q \in \{k+1, k+2, .., p\}$. Let the total number of such projects be X. Thus,

$$X = X_{k+1} + X_{k+2} + ... + X_p$$

This means that there are at least X firms in $C(J_{k+1},R)$. Now, since $|J_k^{\mu}|$ projects are matched under μ, it must be the case that there are at least $X + |J_k^{\mu}|$ firms in $C(J_k,R)$. However, only $X + |J_k^{\mu'}| < X + |J_k^{\mu}|$ firms from $C(J_k,R)$ are matched to projects from $J_{\geq k}$ under μ'. Therefore, there is a firm $f \in C(J_k,R)$ who is either unmatched or is matched to a project in $J_{<k}$ (less preferred to J_k) under μ'. Also, $|J_k^{\mu}| > |J_k^{\mu'}|$ implies there is at least one project $j \in J_k$ which is unmatched under μ'. Then pair (j,f) blocks μ', a contradiction. ∎

This result yields an important insight into the refugee matching problem. Since J_k denotes the set of refugee families of a given size k, the above proposition essentially implies that when the designer insists that the matching satisfy a basic incentive compatibility criterion like stability, there is no room for further increasing the sum total of all refugees who are settled.

2.3 Stable and efficient matchings

In the previous section, we have seen that stable matchings have the same size. This implies that, in the space of stable and Pareto efficient matchings, the designer is not constrained by welfare considerations regarding the matching size. In this section, stability and efficiency are therefore the focus of the study. We propose and discuss various candidate assignment rules to solve any instance of the given problem.

Let us recall that the exogenously given order over the indifference classes of projects $\{J_1, J_2, ..., J_p\}$ is given by: $J_p \succ J_{p-1} \succ ... \succ J_1$.

The assignment rules we discuss are parameterized by a priority order over projects. A **priority order (or, queue order)** over the set of projects J is a bijection mapping $\sigma : J \to J$. To refer to positions in a queue order, we will use the phrase *queue slots*.

Since preferences of one side (firms) are aligned (see, Section 2.2.1), a natural starting point is the priority rule (or, the serial dictator rule), where the other side (projects) is ordered in a queue and allowed to choose a firm in turn. Next, we define this rule formally. Recall that $C(J_k, R)$ denotes the set of firms that are compatible with projects in J_k at preference profile R.

For any given subset of firms $\bar{F} \subset F$, a preference profile R, and a project j such that $j \in J_k$, let

$$B_j(\bar{F} \mid \mathrm{R}) = \begin{cases} \{f \in \bar{F} \cap C(J_k, R) \mid f R_j f' \;\; \forall f' \in \bar{F} \cap C(J_k, R)\} & , \text{if } \bar{F} \cap C(J_k, R) \neq \emptyset \\ \{j\} & , \text{otherwise} \end{cases}$$

denote, the subset of $\bar{F}$ consisting of **best compatible firms** for project j among all compatible firms in $\bar{F}$; if every firm in $\bar{F}$ is incompatible with project j, it is just the set containing project j. Note that the cardinality of $B_j(\bar{F})$ can be strictly greater than one.

Definition 2.3.1. *The **priority rule** is parameterized by an arbitrary priority order σ over projects. For any fixed priority order σ, the priority rule is a correspondence $\psi_\sigma^P : \mathscr{R} \twoheadrightarrow \mathscr{M}$ such that, for any preference profile R we have that $\psi_\sigma^P(R) = M$, with M denoting a set of matchings, where every matching $\mu \in M$ is constructed iteratively such that:*

$$\begin{aligned}
\mu(\sigma(1)) &\in B_{\sigma(1)}\big(F \mid R\big) \\
\mu(\sigma(2)) &\in B_{\sigma(2)}\big(F \setminus \{\mu(\sigma(1))\} \mid R\big) \\
\mu(\sigma(3)) &\in B_{\sigma(3)}\big(F \setminus \{\mu(\sigma(1)), \mu(\sigma(2))\} \mid R\big) \\
&\vdots \\
\mu(\sigma(|J|)) &\in B_{\sigma(|J|)}\big(F \setminus \{\mu(\sigma(1)), \mu(\sigma(2)), ..., \mu(\sigma(|J|-1))\} \mid R\big)
\end{aligned}$$

In words, the priority rule sequentially (according to priority order σ) assigns every project to one of his best compatible firms at every iteration, from the set of remaining compatible firms at that iteration. If no compatible firm is available, then the project remains unmatched. Since preferences exhibit indifferences, based on the choice made at each iteration, several different matchings may be reached by this rule for any fixed profile of preferences.

Given any profile R, an outcome of the priority rule, i.e., a matching $\mu \in \psi_\sigma^P(R)$ for some priority order σ, is not necessarily stable. This is because the priority order σ is not necessarily aligned with the exogenous order $\succ$ over the indifference classes of projects. That is to say, for two projects $j \in J_k$ and $j' \in J_{<k}$[9], it may be the case that $\sigma(j') < \sigma(j)$, i.e., project j' is assigned earlier. Therefore, if project

[9]Recall that, $J_{<k}$ is a shorthand to denote J_x for some $x < k$.

j' is assigned to some firm $f \in C(J_k, R)$ that project j strictly prefers to his own partner, then (j, f) forms a strong blocking pair since firm f is also compatible with project j and hence prefers project j over j'.

The above observation necessitates that the priority order be aligned with the exogenous order $J_p \succ J_{p-1} \succ \ldots \succ J_1$. Accordingly, we define next the notion of a decreasing priority order and correspondingly, the decreasing priority rule. Indeed, it turns out that this new rule solves the above-mentioned issue and always outputs a stable matching (see Proposition 2.3.1).

Definition 2.3.2. *For any priority order σ, the associated decreasing priority order, denoted by σ^D, is evaluated in the following way.*

For any two projects i and j, let $i \in J_x$ and $j \in J_y$. Then σ^D is derived from σ such that:

* *if* $\quad x > y, \quad$ *then* $\quad \sigma^D(i) < \sigma^D(j)$
* *if* $\quad x = y, \quad$ *then* $\quad \sigma^D(i) < \sigma^D(j) \iff \sigma(i) < \sigma(j)$

*A priority order σ is called a **decreasing priority order** if $\sigma = \sigma^D$.*

In words, for any given priority order σ, the associated decreasing priority order σ^D aligns the priority of projects according to the exogenous order $\succ$ over the indifference classes of projects: any project $j \in J_k$ gets a better priority in σ^D over all projects $j' \in J_{<k}$; while, if two projects $j, j' \in J_k$, i.e., they belong to the same indifference class, then their relative priority in σ^D remains the same as in the original priority order σ.

Definition 2.3.3. *The **decreasing priority rule** ψ_σ^{DP} with respect to any fixed priority order σ, is simply the priority rule with respect to the associated decreasing priority order σ^D. That is, for any preference profile R, we have,*

$$\psi_\sigma^{DP}(R) \;=\; \psi_{\sigma^D}^{P}(R)$$

Proposition 2.3.1. *Consider a preference profile $R \in \mathscr{R}$ and let $\mu \in \psi_{\sigma}^{DP}(R)$, then μ is stable but not necessarily efficient.*

The proof of stability appears in Appendix B. Here we state the intuition: since the priority order over projects is aligned according to $\succ$, whenever a firm f strictly prefers a project j over her own partner, it must be the case that this firm f was available to be assigned to project j, which in turn implies project j either prefers his current partner over firm f or is indifferent between them. As a result, the blocking pair of the kind discussed above in the context of the priority rule never forms.

Although always stable, the output (matching) produced by the decreasing priority rule is not necessarily efficient. This inefficiency stems from the combination of two facts: at any given iteration; firstly, the rule does not take into account the preferences of projects that are to be matched in subsequent iterations; and secondly, preferences of projects exhibit indifferences. The following example illustrates this inefficiency.

Example 2.3.1. *Consider a single indifference class $J = \{1,2\}$ of projects and two firms A and B, both of which are compatible with this class.*

Suppose preferences of projects are as below:

R_1	R_2
A , B	A
	B

Then,

$$\psi_{\sigma}^{DP}(R) = \left\{ \begin{pmatrix} 1 & 2 \\ A & B \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ B & A \end{pmatrix} \right\} \text{ for the priority order } \sigma : \sigma(1) = 1, \sigma(2) = 2.$$

It is easy to check that both these matchings are stable, while only the second is also efficient.

In the above example, project 1 is indifferent between firm A and B, and project 1 has the higher priority. Just ensuring that this project gets one of his best compatible firms without considering that project 2 strictly prefers firm A may lead to inefficiency. Additional refinements or adjustments are therefore necessary to guarantee efficiency. Specifically, in any iteration, if one encounters the scenario that there are multiple best compatible firms for the project in question, one must find a way to not outright eliminate any possibility that does not hurt this concerned project. Doing so allows for utility gains to be had in future iterations.

Accordingly, unlike in the case of the priority rule, where we start from the set of all firms and remove a firm at every iteration; we instead modify the procedure and start from the set of all matchings. At any iteration, it is then possible to keep every option (matching) where the currently assigned project receives one of his best compatible firms (even if there are multiple such firms), allowing for full utility gains to be had in future iterations. We call this modified procedure the *refined priority rule*.

To formally define it, we will use the notation $B^*(\cdot\,|\,\mathrm{R})$, which is the counterpart of $B(\cdot\,|\,\mathrm{R})$ presented earlier in the context of the priority rule. The difference is that, B operates on a set of firms while B^* operates on a set of matchings. We will additionally require the following notation. Given a set of matchings $M \subset \mathscr{M}$, a preference profile R, and a project j such that $j \in J_k$, let us denote by $C^*_j(M,R) := \{\mu \in M \mid \mu(j) \in C(J_k,R)\}$, the subset of M where project j is matched to a firm which is compatible with project j at preference profile R.

For any given set of matchings $M \subset \mathscr{M}$, a preference profile R and a project j such that $j \in J_k$, let

$$B^*_j(M\,|\,\mathrm{R}) = \begin{cases} \{\mu \in C^*_j(M,R) \mid \mu(j)\, R_j\, \mu'(j) \;\; \forall \mu' \in C^*_j(M,R)\} & \text{, if } C^*_j(M,R) \neq \emptyset \\ \{\mu \in M \mid \mu(j) = j\} & \text{, otherwise} \end{cases}$$

denote, the subset of M consisting of the **best compatible matchings** for project j in the sense that project j is matched to one of his best-ranked compatible firms among all matchings in M. If there are no matchings where project j is matched to a compatible firm, $B^*_j(R,M)$ is simply the subset of M consisting of all matchings where project j is unmatched.

Definition 2.3.4. *A **refined priority rule** with respect to any fixed priority order σ, is a correspondence $\psi_\sigma^{RP} : \mathscr{R} \twoheadrightarrow \mathscr{M}$ such that, for any preference profile R, we have that $\psi_\sigma^{RP}(R) = M_{|J|}$, where $M_{|J|}$ is defined as follows. Let $M_0 = \mathscr{M}$ (the set of all possible matchings),*

$$\begin{aligned} M_1 &= B^*_{\sigma(1)}\big(M_0 \,|\, R\big) \\ M_2 &= B^*_{\sigma(2)}\big(M_1 \,|\, R\big) \\ &\vdots \\ M_{|J|} &= B^*_{\sigma(|J|)}\big(M_{|J|-1} \,|\, R\big) \end{aligned}$$

Clearly, the above rule can be multi-valued, but note that a direct consequence of its definition is that all projects are indifferent between all elements (matchings) in $M_{|J|}$.

Definition 2.3.5. *The **decreasing refined priority rule** ψ_σ^{DRP} with respect to any fixed priority order σ, is simply the refined priority rule with respect to the associated decreasing priority order σ^D. That is, for any preference profile R, we have,*

$$\psi_\sigma^{DRP}(R) \;=\; \psi_{\sigma^D}^{RP}(R)$$

Proposition 2.3.2. *Consider a preference profile $R \in \mathscr{R}$ and let $\mu \in \psi_\sigma^{DRP}(R)$, then μ is both stable and efficient at R.*

The formal proof of the proposition appears in Appendix B. Here we present an informal discussion. The decreasing refined priority rule always uses a priority order that is aligned with the exogenous order $J_p \succ J_{p-1} \succ \ldots \succ J_1$. This ensures that any selection (matching) from the set $\psi_\sigma^{DRP}(R)$ is stable[10] at profile R . Since this rule is also refined[11], any selection from $\psi_\sigma^{DRP}(R)$ also happens to be efficient

[10] See discussion preceding Definition 2.3.2.

[11] Refer to the section following Example 2.3.1 for an elaborate discussion.

at profile R. Intuitively, since at every iteration, the rule only eliminates those matchings in which, the project assigned at that iteration does not get one of his best firm conditional on every project in all previous iterations receiving one of their best firm; it leaves no room for increasing welfare further. Any attempt at increasing an agent's welfare must come at the expense of another.

Next, we discuss the structure of matchings that are both stable and efficient. For any given preference profile, we ask, whether or not every stable and efficient matching at that profile always results from one of the earlier defined rules.

Proposition 2.3.3. *There exists a preference profile R, and a matching* μ *that is stable and efficient at R such that* $\mu \notin \psi_{\sigma}^{DRP}(R)$.

Proof. Consider the following example.
Suppose we have indifference classes $J_2 = \{1,2\}$ and $J_1 = \{3\}$ of projects and three firms $F = \{A,B,C\}$ compatible with all projects. Let preferences be as follows:

(a) Projects

R_1	R_2	R_3
A , B	A	B
C	C	A, C
	B	

(b) Firms

R_A	R_B	R_C
1, 2	1, 2	1, 2
3	3	3

The matching $\mu = \begin{pmatrix} 1 & 2 & 3 \\ A & C & B \end{pmatrix}$ is *stable* and *efficient* at R.
The two decreasing priority orders are:

$\sigma : \ \sigma(1) = 1, \sigma(2) = 2, \sigma(3) = 3$

$\bar{\sigma} : \ \bar{\sigma}(1) = 2, \bar{\sigma}(2) = 1, \bar{\sigma}(3) = 3$

It is easy to verify that $\mu \notin \psi_{\sigma}^{DRP}(R)$ and $\mu \notin \psi_{\bar{\sigma}}^{DRP}(R)$. ■

Proposition 2.3.3 implies that stability and efficiency do not characterize the decreasing refined priority rule.

Proposition 2.3.4. *There exists a preference profile R, and a matching* μ *that is stable and efficient at R such that* $\mu \notin \psi_{\sigma}^{RP}(R)$.

Proof. Consider the following example with two projects $J = \{1,2\}$ and two firms $F = \{A,B\}$. Preferences are as under:

(a) Projects

R_1	R_2
A , B	A
	B

(b) Firms

R_A	R_B
1	1
2	2

The matching $\mu = \begin{pmatrix} 1 & 2 \\ A & B \end{pmatrix}$ is stable and efficient at this profile.

But the refined priority rule with both possible priority orders produces the matching $\mu' = \begin{pmatrix} 1 & 2 \\ B & A \end{pmatrix}$. ■

For the example in the preceding proof, μ and μ' are the two stable and efficient matchings at profile R. For priority order σ with $\sigma(1) = 1$ and $\sigma(2) = 2$, we have, $\psi_\sigma^{DP}(R) = \{\mu, \mu'\}$, while $\psi_\sigma^{RP}(R) = \{\mu'\}$. This difference is driven by the fact that the latter operates on the set of matchings while the former on the set of firms. Project 2 strictly prefers μ' to μ while project 1 is indifferent between them. Using the refined priority rule, both μ and μ' survive after the first iteration and μ is thereafter eliminated by project 2. But if project 1 is assigned to $A = \mu'(2)$ in the first iteration, it is no longer available for project 2. Therefore, μ can still be generated by a decreasing priority rule.

Proposition 2.3.5 below shows that every stable and efficient matching can be generated by a decreasing priority rule. This demonstrates that matchings that are both stable and efficient are hierarchical by nature: they can be thought of as a result of projects being arranged in a queue order, with each project receiving one of their best firm in turn, from the set of remaining firms. Efficiency is necessary for this result to hold. Stable but not efficient matchings do not necessarily admit this structure (Example 2.3.2).

Example 2.3.2. *One indifference class $J = \{1,2\}$ of projects. Two firms A and B, both of which are compatible with this class. Preferences of projects are as under:*

R_1	R_2
A	B
B	A

The matching $\mu = \begin{pmatrix} 1 & 2 \\ B & A \end{pmatrix}$ *is stable but not efficient. It is easy to check that there is no priority order* σ *such that* $\mu \in \psi^{P}_{\sigma}(R)$.

Proposition 2.3.5. *Consider a preference profile* $R \in \mathscr{R}$ *and let* μ *be a stable and efficient matching at* R*. Then, there is a priority order* σ *over projects such that* $\mu \in \psi^{DP}_{\sigma}(R)$.

The formal proof of this proposition appears in Appendix B. We briefly outline the proof steps next. Fix any indifference class of projects J_r. Since μ is stable at R, a firm $f \in C(J_r, R)$ is matched to some project in $J_{<r}$ under μ only in the event that all projects in J_r are already matched to some firm in $C(J_r, R)$ that they weakly prefer over this firm f. Thus, in a stable matching, the demand of projects in J_r is catered to before projects in $J_{<r}$; and accordingly projects in J_r receive an earlier queue slot compared to projects in $J_{<r}$. This means the priority order is decreasing. We still need the individual queue slots for projects within every indifference class. Fix the top class J_p. By efficiency, we must have that at least one project in J_p is matched to his best firm under μ. This fact identifies precisely the project(s)[12] which receive the earliest queue slot within projects in J_p. Removing the corresponding matched firms from the set of firms and repeatedly applying the same arguments until we exhaust J_p gives the individual queue slots for all projects within J_p. Finally, repeating the process for J_{p-1}, J_{p-2}, $\cdots$, J_1 in turn, completes the proof.

Note that, even though one can always find a priority order σ such that $\mu \in \psi^{DP}_{\sigma}(R)$ for any matching μ that is stable and efficient at R; every selection (matching) from the set $\psi^{DP}_{\sigma}(R)$, despite always being stable, is not guaranteed to be

[12] If there are multiple such projects, any order between them works.

efficient (see, Proposition 2.3.1 or Remark 2.3.1). Therefore, from a practical implementation standpoint, letting a project choose in turn according to a queue order may not always lead to a desirable outcome. The designer is better off using the decreasing refined priority rule.

Let Σ denote the set of all possible decreasing priority orders over the set of projects. For any profile R, denote by $\mathscr{M}_{SE}(R)$ the set of all stable and efficient matchings at R. Next denote by $\mathscr{M}_{DP}(R) := \cup_{\sigma\in\Sigma} \psi_{\sigma}^{DP}(R)$, the set of all matchings that can be obtained using some decreasing priority rule. Similarly, let $\mathscr{M}_{DRP}(R)$ denote the set of all matchings that can be obtained using some decreasing refined priority rule.

The main findings in the preceding results can be summarised as follows:

Remark 2.3.1. *For any preference profile $R \in \mathscr{R}$,*

$$\mathscr{M}_{DRP}(R) \subseteq \mathscr{M}_{SE}(R) \subseteq \mathscr{M}_{DP}(R)$$

Furthermore, there are profiles where one or both of these inclusions above are strict.

2.4 Strongly stable matchings: existence and structure

Strongly stable matchings (those where there are not even weak blocking pairs, where only one member of the pair strictly benefits) are not guaranteed to exist in our preference domain (see, for instance, the example used in Proposition 2.3.4 which does not admit any strongly stable matching). In fact, the existence of a strongly stable matching is a rare event and implies a very specific structure of both preferences and the strongly stable matchings themselves. We outline this phenomenon using the lemmas presented below.

In what follows we assume that the preference profile admits at least one *strongly stable matching* (henceforth, SSM).

Lemma 2.4.1. *Every SSM is efficient.*

Proof. Let μ be a SSM that can be Pareto improved by matching μ'. There is an agent x (firm or project) who strictly prefers μ' to μ. Hence, x is matched at μ', say to some agent y. Since μ' is Pareto superior to μ, y is at least as well at μ' as at μ. Hence, the pair (x,y) weakly blocks μ, a contradiction to μ being a SSM.

■

Lemma 2.4.2. *In any SSM, for every $k \in \{1,2,\cdots,p\}$, the set of projects J_k is either fully matched or fully unmatched.*

Proof. Fix a SSM μ. Suppose there are two project $i, j \in J_k$, such that i is matched with firm f at μ while j is unmatched. Since firm f is indifferent between projects i and j, the pair (j, f) weakly blocks μ, a contradiction. ■

For any matching μ and a set of projects $\hat{J}$, let $\mu(\hat{J})$ denote the set of firms matched with some project in $\hat{J}$.

Lemma 2.4.3. *Fix a preference profile and a SSM μ. For any $k \in \{1,2,\cdots,p\}$, let F_k denote the set of firms that are not matched with some project in $\cup_{x=k+1}^{p} J_x$ at μ. Then,*

(a) For any matched project $j \in J_k$, firm $\mu(j)$ belongs to the top indifference class of this project j among all compatible firms of project j in the set F_k.

(b) A firm $f \in F_k \setminus \mu(J_k)$ does not belong to the top indifference class (within compatible firms in F_k) of any project $j \in J_k$.

(c) $\mu(J_k) = T_k$, where $T_k \subset F_k$ is the set of firms that are in the top indifference class (within compatible firms in F_k) for at least one project $j \in J_k$.

Proof. Part (a): If project j would strictly prefer some firm $f \in F_k$ to $\mu(j)$, then (j, f) would weakly block μ (firm f cannot strictly prefer its current match to j, since it is not matched to a project of better rank).

Part (b): Any firm $f \in F_k \setminus \mu(J_k)$ compatible with projects in J_k, prefers any project $j \in J_k$ to its match in μ. If this firm f is in the top (within compatible firms in F_k) indifference class of any one such project $j \in J_k$, then project j is indifferent between firms f and $\mu(j)$. Hence (j, f) weakly blocks μ, a contradiction.

Part (c): Take any firm $f \in \mu(J_k)$. Then, by Part (a), we have $f \in T_k$ and so, $\mu(J_k) \subset T_k$. Next, assume for contradiction that $f \in T_k$ but $f \notin \mu(J_k)$. This means $f \in F_k \setminus \mu(J_k)$. Then by Part (b) we have, $f \notin T_k$, a contradiction. Thus, $f \in \mu(J_k)$ and so, $T_k \subset \mu(J_k)$. Hence, $\mu(J_k) = T_k$. ■

Lemma 2.4.4. *Let μ be a SSM at preference profile R. Suppose $k^* \in \{1,2,\cdots,p\}$ is the largest integer such that there is a project $j \in J_{k^*}$ which is matched at μ. Then,*

(a) There does not exist a firm f which is compatible with projects in $J_{>k^*}$ at preference profile R.

(b) J_{k^*} is always fully matched at every SSM at R.

Proof. (a): If such a firm would exist, then that firm would form a weak blocking pair with a project of rank better than k^*.

(b): Since there is a project $j \in J_{k^*}$ which is matched at μ, the firm $\mu(j)$ finds projects in J_{k^*} compatible. By Lemma 2.4.3(a), this firm $\mu(j)$ belongs to the top indifference class of project j among all his compatible firms. Therefore, this firm $\mu(j)$ must be matched to a project in J_{k^*} (possibly different from j) in every other SSM. This implies at least one project in J_{k^*} must be matched at every SSM at R. By Lemma 2.4.2, J_{k^*} must then be fully matched in every SSM at R. ■

Lemma 2.4.5. *Let μ and μ' be two SSMs for a given preference profile R. Then the set of firms matched to every indifference class of projects J_k remains the same across these two SSMs, i.e.,*

$$\mu(J_k) = \mu'(J_k) \quad \forall k \in \{1,2,\cdots,p\}.$$

Proof. Let $k^* \in \{1,2,\cdots,p\}$ be the largest threshold such that projects in J_{k^*} are compatible for some firm in F at preference profile R. Then by individual rationality, all projects in $J_{>k^*}$ are unmatched in both μ' and μ. Thus,

$\mu(J_k) = \mu'(J_k)\ \forall k \in \{k^*+1, k^*+2, \cdots, p\}$. Moreover, Lemma 2.4.4 implies that both μ' and μ must fully match all projects in J_{k^*}. By Lemma 2.4.3(c) we have that, $\mu(J_{k^*}) = T_{k^*} = \mu'(J_{k^*})$, where T_{k^*} is the set of all firms which are in the top indifference class for at least one project in J_{k^*}.

Next let $k^{**} \in \{1, 2, \cdots, k^*-1\}$ be the largest threshold such that projects in $J_{k^{**}}$ are compatible for some firm in $F \setminus T_{k^*}$ at preference profile R. If $k^{**} < k^*-1$, then projects in $J_{k^{**}+1}$, $J_{k^{**}+2}, \ldots$, J_{k^*-1} are unmatched in both μ and μ', since there are no compatible firms for these projects at preference profile R who are not already matched to some better-ranked project in both μ and μ'. Furthermore, by Lemma 2.4.4 and Lemma 2.4.3(c) we have $\mu(J_{k^{**}}) = T_{k^{**}} = \mu'(J_{k^{**}})$.

Proceeding in this fashion completes the proof. ■

Lemma 2.4.6. *If a preference profile admits more than one SSM, all projects are indifferent between them.*

Proof. Follows from Lemma 2.4.3(a) and Lemma 2.4.5. ■

The implications from the preceding lemmas can be summarized as follows. If a preference profile admits at least one SSM, then in every such SSM, any set of projects J_k of a given rank is either fully matched or fully unmatched. Moreover, in every such SSM, projects in J_k are matched to the same set of firms (and those matches (firms) are the best for these projects among all compatible firms who are not matched with a better-ranked project). These observations drive the equivalence result in Proposition 2.4.1: if a preference profile admits at least one strongly stable matching, then the set of all strongly stable matchings at that profile, is exactly equal to the set of all matchings that can be generated by a decreasing refined priority rule at that profile.

For any preference profile R, let $\mathscr{M}_{SS}(R)$ denote the set of all matchings that are strongly stable at profile R. Also recall that $\mathscr{M}_{DRP}(R)$ denote the set of all matchings which can be obtained using some decreasing refined priority rule, i.e., $\mathscr{M}_{DRP}(R) = \cup_{\sigma \in \Sigma}\ \psi_{\sigma}^{DRP}(R)$, where Σ denotes the set of all possible decreasing priority order. The following proposition establishes the relationship between these two sets.

Proposition 2.4.1. *Assume that preference profile R admits a SSM. Then we have,*

$$\mathscr{M}_{SS}(R) = \mathscr{M}_{DRP}(R)$$

The proof appears in Appendix B.

2.5 Discussion and concluding remarks

This chapter proposes a domain where the preferences of one side of the market are derived from a master preference list, while agents on the other side have general preferences. The domain permits indifferences. It is shown that stable matchings have the same size. An assignment rule (the decreasing refined priority rule) is proposed that always generates a stable and efficient outcome. In particular, our discussion on the assignment rules demonstrates the hierarchical nature of stable and efficient matchings for the problem. We have also briefly discussed the existence and structure of strongly stable matchings. Here we make some additional remarks about our results.

Firstly, we have not analyzed the strategic aspects of our assignment rules. It is easy to check that the decreasing refined (and, non-refined) priority rule is not manipulable by an individual project. This is not surprising given that the serial dictatorship rule is well known to be strategy-proof.

Secondly, note that the decreasing refined priority rule is quite expensive in terms of the information it must elicit about every project's preference over the entire set of firms. This may become a major concern as the number of agents gets very large. It is in regard to this observation that Proposition 2.3.5 has an important implication which we discuss next.

We know from Proposition 2.3.5 that, for any matching μ that is both stable and efficient at a preference profile R, there is a (decreasing) priority order σ over projects and a (weakly) decreasing sequence $\{\Gamma_j\}_{j=1}^{|J|}$ of choice sets consisting of firms, such that $\mu(\sigma(j)) \in B_{\sigma(j)}(\Gamma_j \,|\, \mathrm{R})$ for all $j \in J$. We conjecture the existence of a sophisticated procedure whereby a stable and efficient matching may be obtained by a recursively defined procedure, where at each step and given the data from previous steps, a choice set for every "project" is defined. The "project" then only needs to report the maximal elements from the choice set. The advantage of such a procedure would be that the projects will no longer need to report the entire preference over the complete list of firms. This can reduce the information elicitation burden greatly and hence can be of particular relevance from a practical point of view where the number of agents becomes significantly large. We leave the study of the existence of such a rule open for future investigation.

2.6 Appendix B - Proofs

Proof Proposition 2.3.1

Fix a preference profile, $R \in \mathscr{R}$ and let $\mu \in \psi_\sigma^{DP}(R)$ for some decreasing priority order σ. Suppose μ is not stable: some project j and some firm f block. This implies project j and firm f form a compatible pair. Let project $j \in J_k$: then firm $f \in C(J_k, R)$. Note that, it cannot be the case that both j and f are unmatched at μ as the rule requires that a project is always assigned a firm if a compatible firm is available. There are three possible remaining cases:

1. $\mu(j) = j$, $\mu(f) = j'$: only firm f is matched.
2. $\mu(j) = f'$, $\mu(f) = f$: only project j is matched.
3. $\mu(j) = f'$, $\mu(f) = j'$: both project j and firm f are matched.

Case:1 – Since (j, f) block we have: $jP_f j'$ which implies $j' \in J_{<k}$. Since ordering is decreasing, this means $\sigma(j) < \sigma(j')$. Since project j comes ahead of project j' in the ordering and remains unmatched, it must be the case that all firms compatible with j were already matched to some project which was ahead in the queue compared to j. This means, every firm in $C(J_k, R)$ which is matched under μ, is matched to some project in $J_{\geq k}$. In particular also firm f. But then $\mu(f) = j' \in J_{<k}$, a contradiction.

Case:2 – Since a firm once matched never gets unmatched, firm f was available to project j when it was assigned f'. This implies $\mu(j) = f' R_j f$. But since (j, f) block we have: $fP_j f'$, which is not possible.

Case:3 – Since (j, f) block we have: $jP_f j'$ which implies $j' \in J_{<k}$, which implies $\sigma(j) < \sigma(j')$. Now, since project j comes ahead in the queue but $\mu(f) = j'$, it means that firm f was available when project j was assigned firm f'. This means that $f' R_j f$. But since (j, f) block we have: $fP_j f'$, which is not possible. ■

Proof of Proposition 2.3.2

Fix a preference profile $R \in \mathscr{R}$ and let $\mu \in \psi_{\sigma}^{DRP}(R)$. Stability of μ follows from arguments in the same vein as in the proof of Proposition 2.3.1 with very minor modifications. We omit it here for the sake of brevity. We proceed to show that μ is efficient. The only way to increase the utility of a project j is to expand the set $M_{\sigma(j)-1}$ of feasible matchings to choose from. This is not possible without decreasing the utility for some project j' such that $\sigma(j') < \sigma(j)$. This is because, if all projects ahead of j in the queue were utility invariant when this welfare improving matching (after carrying out the reshuffling steps) was added, they would not have removed it in the first place. In particular, all (un)matched projects in μ must remain (un)matched after any reshuffling of agents to improve the welfare of any firm. Consequently, such a firm (whose welfare is possibly improved) must have been matched at μ. Also, any such reshuffling attempts requires that the matched projects remain indifferent between partners, before and after the swaps take place.

Suppose next that firm f can be made better without hurting any other agent. Let k^* be the largest integer such that $f \in C(J_{k^*}, R)$, that is: firm f's best ranked acceptable projects are J_{k^*}. Since firm f gets a utility kick, it must mean that f was matched to a project $j \in J_{<k^*}$. Let $\mu(f) = j \in J_{k_j}$. Suppose in the new matching, f gets project $j' \in J_{k_{j'}}$. Then, $k^* \geq k_{j'} > k_j$: j' has a better exogenous rank compared to j. Then firm $\mu(j')$ who becomes partner-less as a result must be assigned a project in $J_{\geq k_{j'}}$. But this project was also matched in μ. Therefore, the firm that then becomes partner-less must also be assigned a project in $J_{\geq k_{j'}}$ and so on. Thus, the Pareto improvement swaps to begin with, must sequentially involve projects in $J_{\geq k_{j'}}$ and firms in $C(J_{k_{j'}}, R)$. Now, let X be the total number of projects in $J_{\geq k_{j'}}$ that were matched in μ. This implies that X firms in $C(J_{k_{j'}}, R)$ were matched to projects with a rank strictly better than project j. Firm f was not one such firm. But since no more than X projects in $J_{\geq k_{j'}}$ can be matched after the reshuffling, there must be a firm in $C(J_{k_{j'}}, R)$ who now is either unmatched or matched to some project in $J_{<k_{j'}}$, a contradiction. ■

Proof of Proposition 2.3.5

Fix a preference profile $R \in \mathscr{R}$ and let μ be a stable and efficient matching. We have to show that there is a (decreasing) priority order σ over projects and a (weakly) decreasing sequence $\{\Gamma_j\}_{j=1}^{|J|}$ of choice sets consisting of firms, such that $\mu(\sigma(j)) \in B_{\sigma(j)}(\Gamma_j)$ for all $j \in J$.

Consider first the top set of projects J_p. If one such project is not matched, but some firm compatible with it (i.e. firm in $C(J_p,R)$) is matched with a project in $J_{<p}$, it violates stability. Thus, either all these projects are matched or all firms compatible with them are matched to a project in J_p, depending on whether $|J_p| \lesseqgtr |C(J_p,R)|$. Now, since μ is efficient, there must exist at least one project in J_p who gets his best firm (one of his best firms if there are many) in $C(J_p,R)$. To see why, suppose not. Next let an arrow go from each project in J_p to all his top choice of firms. Let another set of arrows go from all firms in $C(J_p,R)$ who are matched to some project in J_p, to their corresponding matches. Since the set of agents is finite, there always exists a cycle following arrows. Exchanging along this cycle leads to a Pareto improvement, a contradiction. Collect all such projects who gets one of their best firm under μ in the set X_p^1. Next choose a bijection $\sigma_p^1 : X_p^1 \to \{1,2,...,|X_p^1|\}$. For any $Y \subset J$, let $\mu(Y)$ denote the set (possibly empty) of firms matched to some project in Y. Next, consider the set $J_p \setminus X_p^1$ of projects and the set $C(J_p,R) \setminus \mu(X_p^1)$ of firms. Since μ is efficient we again get a set X_p^2 of projects who get their top choice in $C(J_p,R) \setminus \mu(X_p^1)$. Again choose a bijection $\sigma_p^2 : X_p^2 \to \{|X_p^1|+1, |X_p^1|+2, ..., |X_p^1|+|X_p^2|\}$. Remove the set of matched agents and proceed in the same fashion until we exhaust J_p. We will have a collection of n_p sets $X_p^1, X_p^2, ..., X_p^{n_p}$ and corresponding orderings $\sigma_p^1, \sigma_p^2, ..., \sigma_p^{n_p}$. If $|J_p| > |C(J_p,R)|$, i.e. there are not enough firms (compatible with J_p), then $X_p^{n_p}$ contains all projects in J_p which are unmatched (in the absence of compatible firms, we slightly abuse the notion of top choice to also mean unmatched). If $|J_p| < |C(J_p,R)|$, all projects are matched, and by stability we have, every project in J_p weakly prefers their matched firm over all compatible firms that are unmatched or matched to projects in $J_{<p}$. This fact, together with the way X_p^i's are defined, we see that every project in J_p is matched to one of their best available compatible firms when they allowed to choose (following the order).

Next, consider the second-best set of projects and carry out the same procedure, followed by the third best and so on. To elaborate, proceeding in decreasing order, for each $k \in \{p-1, p-2, ..., 1\}$, let us consider projects in J_k and firms in $C(J_k, R) \setminus \bigcup_{i=k+1}^{p} \mu(J_i)$. Then carrying out the same steps outlined earlier we will get a collection of n_k sets $\{X_k^i\}_{i=1}^{n_k}$ and corresponding orderings $\{\sigma_k^i\}_{i=1}^{n_k}$. Once we do this for every k, we finally have a collection $\{X_k^i\}_{\substack{k=1(1)p \\ i=1(1)n_k}}$ and $\{\sigma_k^i\}_{\substack{k=1(1)p \\ i=1(1)n_k}}$.

Next, define the priority order σ such that $\sigma(j) = \sigma_k^i(j)$, if $j \in X_k^i$. Note that σ is by construction aligned with the exogenous order over projects: $J_p \succ J_{p-1} \succ ... \succ J_1$.

Finally, defining the collection $\{\Gamma_j\}_{j=1}^{|J|}$ of choice sets consisting of firms such that; $\Gamma_1 = F$, and for $j = \{2, 3, ..., |J|\}$, $\Gamma_j = F \setminus \bigcup_{i=1}^{j-1} \mu(\sigma(i))$, we have the desired proof. ■

Proof of Proposition 2.4.1

We use the lemmas in Section 2.4 including the notations therein. Suppose R is such a preference profile that admits a SSM.

Let $k^* \in \{1, 2, \cdots, p\}$ be the largest threshold such that projects in J_{k^*} are compatible for some firm in F at preference profile R. Then by individual rationality, all projects in $J_{k^*+1} \cup J_{k^*+2} \cup \cdots \cup J_p$ remain unmatched at every SSM at preference profile R. Moreover, all projects in $J_{k^*+1} \cup J_{k^*+2} \cup \cdots \cup J_p$ are also unmatched at every matching that is output by any decreasing refined priority rule by definition.

Since there is a compatible firm for projects J_{k^*} at profile R, at least one such project can be and must be matched. Moreover, since preference profile R admits a SSM, by Lemma 2.4.2,

- All projects in J_{k^*} can be matched.

- All projects in J_{k^*} can be matched to T_{k^*} such that each project is matched to a best compatible firm for that project from within the set F.
- Firms in the set $F \setminus T_{k^*}$ are not in the top indifference class (within compatible firms in F) for any project in J_{k^*}.

This implies that, no matter what ordering of projects within J_{k^*} is used by the decreasing refined priority rule, the rule always outputs a set of matchings that matches J_{k^*} to T_{k^*} (and never to a firm in $F \setminus T_{k^*}$), with every project in J_{k^*} getting one of their best compatible firm in F. Since every SSM at profile R satisfy the above bullet points, the result is true for projects in J_{k^*}.

Note that, the set of matchings from which projects of rank lower than k^* gets to pick from in the decreasing refined priority rule, never contains a matching where this lower-ranked project is matched to a firm in T_{k^*}. Moreover, firms in T_{k^*} never prefer these lower-ranked projects to their SSM match. Next, let $k^{**} \in \{1,2,\cdots,k^*-1\}$ be the largest threshold such that projects in $J_{k^{**}}$ are compatible for some firm in $F \setminus T_{k^*}$ at preference profile R. If $k^{**} < k^*-1$, then all projects in $J_{k^{**}+1} \cup J_{k^{**}+2} \cup \cdots \cup J_{k^*-1}$ remain unmatched at every SSM at preference profile R; as well as at every matching that is output by any decreasing refined priority rule (by definition) at profile R. By the same arguments as above, every SSM and every decreasing refined priority rule at preference profile R, matches projects in $J_{k^{**}}$ to $T_{k^{**}}$ $\big($and never to a firm in $(F \setminus T_{k^*}) \setminus T_{k^{**}}\big)$, with every project in $J_{k^{**}}$ getting one of their best compatible firm in $F \setminus T_{k^*}$. Continuing in this manner we have the desired result. ■

Chapter 3

Dynamic job rotation: an impossibility result

Abstract

Job rotation is the practice of moving employees between jobs in an enterprise. In this chapter, we model the organization of job rotation in a dynamic matching framework. Agents have strict preferences over jobs and jobs have strict priorities over agents. To facilitate job rotation, we construct a job rotation priority structure which entails: if an agent is matched to a job at a given period, then this agent must give it up in the next period should another agent desire it. Our model is the exact opposite of the dynamic school choice problem. We show that constrained efficiency and strategy-proofness, the two most important properties in priority-based allocation problems, are incompatible in the most natural setting of dynamic job rotation.

3.1 Introduction

Job rotation is a well-known organizational development technique, used by employers to move their employees through a range of job functions, in order to boost interest and motivation. The use of such a strategy is widespread across both private and public sectors (Osterman, 1994; Ostrom, 1990; Berkes, 1992).

We model this organization of job rotation in a dynamic matching framework. In particular, this chapter contributes to the literature on dynamic priority-based allocation of indivisible objects such as: allocating teachers to public schools (Pereyra, 2013), or children to daycare facilities (Kennes et al., 2014).

One common theme in the above papers is that, when an agent is allocated an object at some period, owing to individual rationality concerns, this agent enjoys a higher priority for that object in future periods over all other agents. On the contrary, in the present context, it is the exact opposite. To facilitate job rotation, employers require that if an agent is matched to an object (job) at a given period, then this agent must give it up in the next period should another agent desire it.

Stability is a central concept in matching theory. In priority-based allocation problems, which is the subject of this chapter, it embodies a notion of fairness and is often referred to as *justified envy-freeness*: an agent is said to have *justified envy* towards another agent if the latter is assigned an object for which the former has higher priority, and a stable matching eliminates the occurrence of such envy. Our use of stability throughout this chapter is going to strictly adhere to the spirit of *justified envy-freeness*. In dynamic matching environments, the approach in the literature, therefore, has been to appropriately adapt the above concept to the notion of dynamic stability. Respecting these priorities makes the set of feasible matchings smaller, oftentimes conflicting with efficiency concerns. Accordingly, a constrained efficient matching is one that is dynamically stable (therefore, respects priorities) and cannot be Pareto improved without necessarily violating priorities.

In the context of priority-based allocation of objects, strategy-proofness of a mechanism is another desirable property that is frequently studied in conjunction with dynamic stability. It ensures that no individual can gain by reporting false preferences. A strategy-proof mechanism is appealing because it gives straightforward incentives to each individual participant, whether or not this agent has any information regarding the preferences of other agents.

We show that these two important and appealing properties are incompatible in the most natural dynamic job rotation framework: there does not exist a constrained efficient and strategy-proof mechanism.

Our framework consists of a simple two-period model with a fixed set of employees and jobs. All employees and all jobs are present in both periods. For every job, there is an exogenously given strict priority ordering over agents. One can think of these priorities to reflect the suitability of agents (based on skill-set, qualification, past experiences, etc.) for each job function that the employer wants as part of the rotation scheme. Employees have strict preferences over jobs. Employees report time-invariant preferences at the start of the process. These preferences are then extended to time-separable preferences over job allocation tuples. To achieve job rotation, if an employee is matched to a job in the first period, then this agent is artificially moved to the lowest spot in the priority for that job in the second period. This is encoded in what we call the *job rotation priority structure*. We then show that, if one desires to find a mechanism that rotates employees through jobs as efficiently as possible while respecting the job rotation priority structure (i.e., ensuring that there is no justified envy), then one must sacrifice on its incentive property.

Game theoretic study of the job rotation problem is very recent. In a market design framework, Yu and Zhang (2020) analyzes the static version of the problem. They impose that every agent initially occupies a position (job). Job rotation is encoded by weak priorities: all agents not initially matched to the job have equal highest priority for it while the agent who initially occupies the job has the least. They propose an algorithm that always finds a constrained efficient and weakly group strategy-proof matching. Korpela et al. (2021) study the static job rotation problem in the more general setting of implementation theory.

3.2 Model

Time is discrete: starts at $t = 1$ and ends at $t = 2$. The two periods are enough to capture the main forces at play that drive the result.

An instance of a *job rotation problem* consists of:

- A set of agents (or, employees) $I = \{1, 2, 3, \ldots\}$.

- A set of objects (or, job functions) $O = \{a, b, c, \ldots\}$.
- A preference profile $R = (R_i)_{i \in I}$, where R_i is a time-invariant strict preference relation of agent i over jobs in O.
- A priority list $\succ = (\succ_o)_{o \in O}$, where every job o has a strict priority $\succ_o$ over agents in I. The priority $\succ_o$ reflects the suitability of agents (based on skill-set, qualification, past experiences, etc.) for the job function o.

We will interchangeably use the terms "agent" and "employee" in this chapter. We assume throughout that $|I| = |O|$. Let $\mathscr{R}$ denote the set of all strict preference relations on O. Let $\mathscr{P}$ denote the set of all strict priorities over agents in I. For any given I and O, the *job rotation problem* is denoted by $\langle I,\ O,\ \mathscr{P}^{|O|},\ \mathscr{R}^{|I|}\rangle$.

Agents have to be assigned to jobs over these two periods. Agents cannot remain unmatched in any period. The tuple $(a,b) \in O \times O$ denotes a **job allocation plan** over two periods. When confusion is unlikely, we write ab for (a,b).

Preferences of agents over jobs are extended to time-separable preferences over job allocation plans. This is described next.

3.2.1 Preferences over job allocation plans

Preference $R_i \in \mathscr{P}$ over O is extended to preference $\overline{R}_i$ over $O \times O$ in the following way. Let $\overline{P}_i$ denote the asymmetric part of $\overline{R}_i$. For any two job allocation plans (a,b), $(a',b') \in O \times O$,

- if $a R_i a'$ and $b R_i b'$, then $(a,b)\ \overline{R}_i\ (a',b')$
- if $a R_i a'$ and $b R_i b'$ are such that at least one of these statements holds strictly, then $(a,b)\ \overline{P}_i\ (a',b')$

Let $\overline{\mathscr{R}}$ denote the set of all such possible preferences over $O \times O$. It is to be noted that the preference domain $\overline{\mathscr{R}}$ is incomplete.

3.2.2 Matchings

A period$-t$ matching μ^t is a function $\mu^t : I \cup O \to I \cup O$ mapping agents to jobs in period t such that:

- $\mu^t(i) \in O$ for all $i \in I$,
- $\mu^t(o) \in I$ for all $o \in O$, and
- $\mu^t(i) = o \iff \mu^t(o) = i$ for all $i \in I$.

A *matching* μ is a tuple of period matchings, $\mu = (\mu^1,\ \mu^2)$ where μ^1 and μ^2 are period-1 and period-2 matchings. The set of all possible matchings is denoted by $\mathscr{M}$.

3.2.3 Job rotation priority structure

Recall that every job (job function) o is endowed with an exogenously given (strict) priority $\succ_o \in \mathscr{P}$ over I. These priorities can be thought to have been derived from observable employee characteristics such as their skill-set, work experience, etc. For any given job o, and for two employees i and j, suppose we have that $i \succ_o j$. While the employer would like to give employee i preferential access to job o, the employer would also like to encourage rotation. In order to achieve a reasonable compromise, the employer may make use of the temporal dimension: as long as neither employee is assigned to job o, employee i gets priority over j; but if i is assigned to job o in a previous period (and therefore gets experience in the job), then employee j gets a chance (over i) instead to garner some experience in job o, in spite of the fact that j has a lower priority than i for this job to begin with.

Therefore, in order to facilitate rotation, the discussion above calls for the job priorities to be artificially modified based on first-period job assignments. This is encoded in the *job rotation priority structure*, which is defined below. In words, the structure encodes the following: if an agent i is assigned to a job o in the first period, then this agent i necessarily has the least priority for that job o in the second period, while the priorities of all other agents for that job o obey the exogenous order $\succ_o$. Accordingly, the job priorities in the second period therefore depend on the job assignments in effect during the first.

Definition 3.2.1. *Given a matching $\mu = (\mu^1,\ \mu^2) \in \mathscr{M}$ and a priority list $\succ \in \mathscr{P}^{|O|}$, the associated* ***job rotation priority structure*** *is a list $\overline{\succ}(\mu,\succ) := (\overline{\succ}^1_o,\ \overline{\succ}^2_o)_{o\in O}$ where, $\overline{\succ}^t_o$ denotes priority of job o at period t, and is derived as follows:*

- $\overline{\succ}^1_o = \succ_o$ *for all $o \in O$.*
- *For all $o \in O$, if we have $o = \mu^1(i)$ i.e., employee i is assigned to job o at $t = 1$, then $\overline{\succ}^2_o$ is defined as follows:*
  - *for all $j \neq i$ we have $j \ \overline{\succ}^2_o\ i$.*
  - *for all $j,k \neq i$ we have $j \ \overline{\succ}^2_o\ k$ if and only if $j \succ_o k$.*

3.2.4 The criteria for matchings

Definition 3.2.2. *Given a preference profile $R \in \mathscr{R}^{|I|}$ and a priority list $\succ \in \mathscr{P}^{|O|}$, consider a matching $\mu = (\mu^1,\ \mu^2) \in \mathscr{M}$ and its associated job rotation priority structure $\overline{\succ}(\mu,\succ) = (\overline{\succ}^1_o,\ \overline{\succ}^2_o)_{o\in O}$. We say that μ is* ***dynamically stable*** *at $(R,\succ)$, if there does not exist an employee-job pair $(i,o) \in I \times O$ such that one of the following two conditions hold:*

- $(o,\ \mu^2(i))\ \overline{P}_i\ \mu(i)$ *and* $i\ \overline{\succ}^1_o\ \mu^1(o)$
- $(\mu^1(i),\ o)\ \overline{P}_i\ \mu(i)$ *and* $i\ \overline{\succ}^2_o\ \mu^2(o)$

The interpretation of dynamic stability in this chapter is strictly in the spirit of *justified envy-freeness*. Before expanding on it, an important fact to point out first is that the job matching that is to be implemented over the two periods is known to all employees at the start of the job rotation process[1], i.e., every employee knows her two-period job allocation at the start of the process. Now, given a matching, it is inevitable that some employees will envy the job assignment of another. Among

[1]This is because the employer commits to a mechanism at the start of the rotation process. For further details see the discussion on *mechanisms* in the next sub-section.

such cases of envy, the ones in which the employee who envies another has a higher priority for the job in question (and this employee has not been assigned to this job in the previous period), are arguably justified. A dynamically stable matching eliminates any occurrence of such justified envy. Therefore, in the event some employees were to express dissatisfaction/objection regarding a proposed job matching that is dynamically stable, all such objections are guaranteed to be objectively[2] unjustified.

Remark 3.2.1. *Note that dynamic stability does not imply that an agent cannot be assigned to the same job in consecutive periods. This is permitted so long as every other agent is happy with their own assignment.*

Next, we define Pareto efficiency, a very standard notion in economics.

Definition 3.2.3. *Consider a preference profile* $R \in \mathscr{P}^{|I|}$. *A matching* $\bar{\mu} \in \mathscr{M}$ ***Pareto dominates*** *another matching* $\mu \in \mathscr{M}$ *at* R *if,*

$$\bar{\mu}(i)\ \overline{R}_i\ \mu(i) \ \textit{ for all } \ i \in I \quad \textit{and} \quad \bar{\mu}(j)\ \overline{P}_j\ \mu(j) \ \textit{ for some } j \in I$$

A matching $\mu \in \mathscr{M}$ *is* ***Pareto efficient*** *at preference profile* R *if it is not Pareto dominated by any other matching at R.*

It is easy to come up with examples (with two jobs, and two employees) to show that: (i) a dynamically stable matching may be inefficient; (ii) an efficient matching need not be dynamically stable.

Since job rotation is the primary concern for the employer, we seek to first ensure that the matching is dynamically stable. Having met this primary goal, we would like the matching to be as efficient as possible. This motivates the following definition.

Definition 3.2.4. *Given a preference profile* $R \in \mathscr{R}^{|I|}$ *and a priority list* $\succ \in \mathscr{P}^{|O|}$, *a matching* $\mu \in \mathscr{M}$ *is* ***constrained efficient*** *at* $(R, \succ)$ *if it is dynamically stable at* $(R, \succ)$ *and is not Pareto dominated by any other dynamically stable matching.*

[2] See the discussion on *job rotation priority structure* in the previous sub-section for the basis of this objectivity.

3.2.5 Mechanisms

At the outset, the employer collects preference reports from the employees and possesses a list of job priorities. Based on these reports and the priority list, a matching needs to be selected. A mechanism provides a systematic way to select a matching at all possible contingencies, i.e., at every preference and priority profile.

Recall, the set of all possible *matchings* is denoted by $\mathscr{M}$.

A **mechanism** is a function $\Psi : \mathscr{R}^{|I|} \times \mathscr{P}^{|O|} \longrightarrow \mathscr{M}$ that chooses a matching $\Psi(R, \succ) \in \mathscr{M}$ for every $R \in \mathscr{R}^{|I|}$, and any given $\succ \in \mathscr{P}^{|O|}$. For every $i \in I$, let $\Psi_i(R, \succ) \in O \times O$ denote the allocation plan of agent i at $(R, \succ)$.

The employer commits to a mechanism at the start of the job rotation process. Therefore, once the employees report their preferences, the job matching to take effect over the two periods becomes known to everybody. The pertinent question then is: which mechanism should be chosen? We pursue two appealing properties that the employer would like the mechanism to satisfy.

An agent i *manipulates* a mechanism Ψ at $(R, \succ)$ by reporting $R'_i \in \mathscr{R}$ if,

$$\Psi_i((R'_i, R_{-i}), \succ)\ \overline{P}_i\ \Psi_i(R, \succ)$$

A mechanism Ψ is **strategy-proof** if it is never manipulated by any agent. A strategy-proof mechanism induces truth-telling: as long as an employee has no information about what the other employees' reports are, she fares best or at least not worse by being truthful in her own report. A mechanism Ψ is **Pareto efficient** if it always outputs a Pareto efficient matching at every profile. A mechanism Ψ is **constrained efficient** if, for all $R \in \mathscr{R}^{|I|}$ and for all $\succ \in \mathscr{P}^{|O|}$, the matching $\Psi(R, \succ)$ is constrained efficient at $(R, \succ)$.

3.3 The impossibility result

Theorem 3.3.1. *Consider any job rotation problem* $\langle I, O, \mathscr{P}^{|O|}, \mathscr{R}^{|I|}\rangle$ *with* $|I| \geq 5$. *Then, there does not exist a constrained efficient and strategy-proof mechanism.*

Proof. Consider a problem where there are five agents and five jobs as below: $I = \{1,2,3,4,5\}$ and $O = \{a,b,c,d,e\}$.

Next, consider two preference profiles $R = (R_1, R_2, R_3, R_4, R_5)$ and $R' = (R'_1, R_2, R_3, R_4, R_5)$, and a priority list $\succ$ as presented in Table 3.1.

R_1	R_2	R_3	R_4	R_5
c	a	b	d	d
e	b	a	a	e
	e	c		c

(a) R

R'_1	R_2	R_3	R_4	R_5
a	a	b	d	d
c	b	a	a	e
e	e	c		c

(b) R'

$\succ_a$	$\succ_b$	$\succ_c$	$\succ_d$	$\succ_e$
4	2	1	4	5
3	3		5	2
1	1			
2				
5				

(c) $\succ$

Table 3.1: Preference profiles and priority structure for the proof of Theorem 3.3.1.

We first show that there is a unique *constrained efficient* matching μ at R given by:

$$\mu|_{R=(R_1, R_2, R_3, R_4, R_5)} = \begin{pmatrix} 1 & : & ce \\ 2 & : & ab \\ 3 & : & bc \\ 4 & : & da \\ 5 & : & ed \end{pmatrix}$$

where, the notation $1 : ce$ is read as "agent 1 is matched to job c and e in the first and second period respectively", and so on.

To see this, note that given any period-2 matching μ^2:

- stability requires that 4 gets d in the first period as it is the best job for 4 and also has the highest priority for it.
- Similarly, by stability 1 must get c.

- Given 4 gets d, stability then requires 5 gets job e since 5 has the highest priority for e.

- Now, agents 2 and 3 are yet to be assigned. The remaining jobs are a and b. Note that, both the possible matchings involving 2 and 3 are stable. However, the matching where 2 gets a and 3 gets b Pareto dominates the other matching.

Therefore, irrespective of the second-period matching produced, any *constrained efficient* mechanism must have the following first period matching μ^1 at preference profile $R = (R_1, R_2, R_3, R_4, R_5)$

$$\mu^1(1,2,3,4,5) = (c,a,b,d,e)$$

Next, we show that any constrained efficient mechanism must produce the matching $\mu^2(1,2,3,4,5) = (e,b,c,a,d)$ in the second period.

To see this, note that given period-1 matching μ^1:

- Agent 4 now has the least priority for job d while agent 5 now has the highest. Since d is the best job for 5, stability implies that 5 gets d.

- Given that, stability implies 4 gets a as she has the highest priority for a.

- If 4 gets a, then stability implies 2 must get b since agent 2 has the highest priority for b in the second period.

- After a and b have been allocated, job c is the best available job for agent 3. Since agent 1 has the least priority for c in the second period, agent 3 must get c.

- Lastly, agent 1 therefore gets e.

In the same spirit, we can show that there is a unique *stable*, and thereby *constrained efficient* matching μ also at R'

$$\mu|_{R'=(R'_1, R_2, R_3, R_4, R_5)} = \begin{pmatrix} 1 & : & cc \\ 2 & : & be \\ 3 & : & ab \\ 4 & : & da \\ 5 & : & ed \end{pmatrix}$$

Therefore, any constrained efficient mechanism must produce the above matchings at profiles R and R' for the given priority structure. But this means, agent 1 at R_1 can manipulate by reporting R'_1 and do strictly better. Consequently, the mechanism fails to be strategy-proof. ■

Since dynamic-stability and Pareto efficiency implies constrained efficiency, we have the corollary below.

A mechanism Ψ is said to be **dynamically stable** if, for all $R \in \mathscr{R}^{|I|}$ and for all $\succ \in \mathscr{P}^{|O|}$, the matching $\Psi(R, \succ)$ is dynamically stable at $(R, \succ)$.

Corollary 3.3.1. *Consider any job rotation problem $\langle I, O, \mathscr{P}^{|O|}, \mathscr{R}^{|I|} \rangle$ with $|I| \geq 5$. Then, there does not exist a dynamically stable, Pareto efficient, and strategy-proof mechanism.*

The main driving force behind the impossibility result is discussed next. It is well known for the static problem that the DA mechanism is strategy-proof for the proposing side (Dubins and Freedman, 1981). Therefore, if one were to run DA for allocating jobs in the first period, no agents could improve their first-period assignment. However, the counterexample in the above proof shows that it is possible for an agent to change the first-period assignment of another agent without changing her own. But, since first-period assignments influence the job priority structure (specifically, the job priorities next period), there are situations (as highlighted by the example), where this manipulation may be profitable in improving her second-period allocation.

3.4 Conclusion and further direction

In this chapter, we have studied the organization of job rotation in the most natural dynamic matching framework. In a very simple two-period setup, we have shown that it is not possible to find a mechanism that is strategy-proof and constrained efficient: the two most important and well-studied properties in the context of priority-based assignment of indivisible objects.

The preference domain used to model employees' time-invariant preferences is incomplete. Since we prove a negative result, our result remains true for any arbitrary resolution of this incompleteness. To elaborate, consider an agent i and job allocation plans (a,b) and (b,a). Suppose agent i strictly prefers a to b. In our preference domain, the plans (a,b) and (b,a) are incomparable. But irrespective of how this is resolved, the impossibility holds. Furthermore, the result remains true in two other preference domains[3]:

1. Domain where agents evaluate two job allocation plans by comparing the worse job in one plan against that of the other.

2. Domain where agents evaluate two job allocation plans by comparing the total utility obtained from each plan, where the total utility from any given plan is derived as follows. Consider an agent i, and fix any time-invariant strict preference R_i over jobs. Assign a score of $|O|$ to the best-ranked job according to R_i, a score of $|O|-1$ to the second-best ranked job, and so on[4]. Then, the utility from a job allocation plan (a,b) is simply the sum of scores of a and b according to R_i.

These two preference domains along with ours, we believe, are the most natural ways to think about extending preferences over jobs to preferences over job allocation plans. The properties of mechanisms that we have studied are the most desirable ones as well. Given this, the message of this chapter then is that the

[3]The same example used in the proof of the impossibility theorem work for these two domains.

[4]Recall that O denotes the set of jobs.

"first best" solution to the dynamic rotation problem is not possible. It is therefore worthwhile for future work to investigate ways of escaping this impossibility: one direction where we conjecture a positive result is by relaxing the assumption of strict job priorities, and allowing instead that all employees not already matched to a given job in a previous period, have equal access to that job. Such an assumption is reasonable when (i) the jobs in question don't require a high level of skill and prior experience; or (ii) the pool of employees is homogeneous with respect to skill-set and experience. One other way to escape the impossibility might perhaps be to weaken the strategy-proofness notion in a clever way. We leave these directions for further research.

This page is intentionally left blank.

Part II

Contest Theory

Chapter 4

Effort dynamics in a competitive league

Abstract

In this chapter, we study the problem of a team manager in a competitive league, where the manager is required to make decisions about the overall *effort (energy)* level that the team exerts in each game, across the whole season. In particular, we focus on effort dynamics "late" in the game and the impact of *score-difference* up to that point in the game, on the effort choice. Central to the analysis is the tension between "saving energy for future games" and "winning the current game" in a long season. We model decisions as a Markov Decision Process, and solve the model computationally to find that, indeed, saving energy is optimal for the team manager throughout most of the season. This conclusion is robust to several variations of the base model we consider, including against a field of teams whose coaches employ similar strategies.

4.1 Introduction

The final minutes of a basketball game are when legends are made or hearts are broken. It is what Michael Jordan passionately deemed "Winning Time". Coaching decisions made during this period are critical to a team's chances of success. It is also the most heated topic of discussion among fans. In the course of sports leagues, usually lasting several months, it is expected that the abilities of teams taking part in the season (or, tournament) will change over time. Therefore, it is not hard to fathom why a coach would be inclined to hold back or rest his players in a particular game in consideration of a farsighted objective — maximizing the success rate of the team in the whole season. Several considerations may play at the back of his mind — take for example: the current fatigue level of the team, strength of opponent, current score in the game, energy level of star players, schedule of upcoming games, etc. Economics research on relative performance evaluation has mainly focused on the comparison of final performances between competitors. In this article, we instead look at effort dynamics "late" in a game and the impact of information feedback (score-difference) up to that point in the game, on the effort choice thereafter. We ask: when is it optimal, if at all, for the coach to save energy for late in the current game in consideration of long-term team success?

To this end, we consider the problem of a coach (or, team manager; or, even the players) in a competitive league, where they are required to make decisions about the overall effort level that the team exerts in each game, across the whole season. The sports we have in mind are ones such as basketball, volleyball, handball, etc., where the score-difference between competing teams can change by large margins very rapidly and frequently. We will use the basketball terminology in the remainder of the article. The sequence of opponents that the team manager (or, coach) faces over the season, is collectively called the *field*, as is common, e.g., in evolutionary game theory.

During a game, players are exposed to frequent high-intensity movements like sprinting, jumping, acceleration, deceleration, and abrupt changes of direction; which can lead to acute and accumulated fatigue. Fatigue may affect the ability of the players to perform over the course of a lengthy season. Oftentimes, several games are played on consecutive days. These factors render the need for not only monitoring fatigue level (see, for example, Thorpe et al. (2017), Stojanović et al. (2018)); but more generally warrants an effective management of the team energy level. This is particularly more relevant for the case of star players. Resting or limiting minutes of star players is indeed something coaches frequently consider.

Therefore, it is no surprise that the coach may want to hold back the intensity of the team (or limit the minutes of his star players) early in the game, and make an informed scientific decision based on the scores at a later point in the game. However, not exerting sufficient effort may prove to be costly when it comes to winning the game. Our model explores this "saving energy for future games" versus "winning the current game" dilemma in a long season.

The model is formally presented in Section 4.2. The setting, in a nutshell, is as follows: at the start of each *game*, effort (energy) reserves are realized for both the *team* and the *field*. Score-difference at the end of the third quarter (*advantage*) is probabilistically determined as a function of effort exerted by both teams during those quarters. At the beginning of the fourth quarter (Q4), the coach has to decide between how much more effort (energy) to have their players exert in the current game; and how much to save for future games; as a function of the *advantage* at the beginning of Q4. Effort reserves for the *team* and the *field* are replenished (stochastically) before the start of the next game. The coach's goal is to maximize the expected sum of winning in the whole season.

We solve the above model (henceforth, base model) computationally making specific choices for the probability distributions governing the process. In the base model, the *field* is assumed to split its energy equally in all quarters of every game, i.e., it does not act as a strategic agent, therefore making the base model a single-player decision problem. As one would expect, we show that initial conditions are inconsequential in a long season (Proposition 4.2.2). The *Expected Win Rate* (*EWR*), which is defined as "the average expected value (winning probability) in games remaining", and the optimal strategies are presented in Section 4.3. The *EWR* at different stages of the season is consistent with the predictions from Proposition 4.2.2 (Figure 4.1). The results also show that, indeed, the manager finds it optimal to save energy throughout most of the season (Figure 4.2). This conclusion from the base model is robust to several extensions that we consider which are outlined below.

As a first extension, we consider the case where the managers of the *field* of teams employ similar strategies (see, Section 4.4.1). Very similar conclusions hold here. Furthermore, we show in Section 4.4.1 that the optimal strategy (evaluated when the *field* plays strategically), if played by both the *team* and the *field*, constitutes an approximate equilibrium of the game.

The second extension challenges the assumption made in the base model that an exogenously given constant fraction α of the effort reserve is used up by the *team*, throughout the first three quarters combined. In Section 4.4.2, we relax this assumption and allow the manager to act strategically during the first three quarters ($Q1$ through to $Q3$ combined) as well. We find interestingly that the energy used in the first three quarters peaks at about 75%. This is not obvious to begin with — one might think perhaps to put in a greater effort in the first three quarters to get a large lead. However, this is not the case as we see from the optimal strategies (Figure 4.10b). Allowing the coach to have greater control can indeed be helpful (we see the EWR is higher than in the base model; see, Section 4.4.4) because it allows the manager to decide when facing a team that has high energy, if it may wish to "not bother" and save its energy instead.

The final extension in Section 4.4.3 is motivated by the observation that participating teams can usually be broadly classified into two types: *strong* and *weak*; and that the schedule of upcoming games is usually known to coaches in advance. The above classification may be based on factors such as the overall team strength; or team ranking as published by a recognized sporting association or governing authority (e.g., FIBA for basketball, World Rugby for rugby union); or record of recent team performances, etc. The coach, therefore, has finer information regarding what type of opponent to expect in the upcoming game, however, the exact energy reserve that the opponent comes with on match day is still unknown to the coach today, i.e., at the time they are contemplating the crucial saving decision that must be made in the ongoing game.

Finally, in Section 4.4.4 we summarize and present a detailed comparison of EWR and optimal strategies in each of the extensions with respect to the base model.

4.1.1 Related literature

Individuals, groups, and teams who are behind their opponents in competition tend to be more likely to lose. Berger and Pope (2011) in contrast show that through increasing motivation, being slightly behind can actually increase success. They study whether a team losing at halftime is more likely to win than expected using a logit model. They find that usually the higher the score difference the more likely the team is to win. But if the halftime score difference is small they observe a discontinuity - being behind with a small difference (e.g. down by 1 point) can lead to, an increase in effort, and a win in the game. Thus they provide evidence that being behind might lead to winning.

Several papers study situational variables (such as location, opposition strength, or game status) that might influence performance (Gómez et al. (2013), Sampaio et al. (2010) , O'Donoghue (2009)). The quality of opponents is particularly important and is usually addressed by categorizing teams as "strong" or "weak" (Sampaio et al. (2013), Lago (2009), Taylor et al. (2008)). Several applied statistics papers have considered dynamic developments of abilities (see, for example, Cattelan et al. (2013), Rue and Salvesen (2000), Crowder et al. (2002)).

Other observations in competitive leagues have been made in the literature: Arkes and Martinez (2011) develop an econometric model to determine if there is a momentum effect (an effect of success in the past few games, over and above the effect of team quality) in the NBA by examining how success over the past few games leads to a higher probability of winning the next game. They find strong evidence for a positive momentum effect. Neiman and Loewenstein (2011) study reinforcement learning in professional basketball. They show that players substantially change their behaviour, manifested as their rate of 3pt shots, in response to the outcome of a single 3pt. Moreover, this change is associated with decreased performance, as measured by 3pt percentage and 3pt return. Their results indicate that despite years of experience and high motivation, professional players overgeneralize from the outcomes of their most recent shots, which leads to decreased performance.

The trade-off between "saving energy" and "winning current game" is central in our model. Thus it is related to the literature on the optimal consumption-investment problem facing a utility-maximizing agent (an individual or a household) that is subject to bankruptcy, the utility being associated with consumption and with bankruptcy (classical models include: Samuelson (1969), Merton (1971); other extensions and applications: Kraft and Munk (2011), Geyer et al. (2009), Zariphopoulou (1994)).

Dynamic contest models with budget constraints bear resemblance to the stage game in our model. In particular, models of dynamic Colonel Blotto games. Players start with some resource budget which must be split into a series of contests. At each stage (contest), they must decide how much of their remaining budgets to invest in trying to win the current battle ("quarter" in our model), and how much to save for possible later battles. Different winning objectives like "winning majority of the contests" or "maximizing number of wins" have been studied in the literature (see, for instance, Friedman (1958), Sela and Erez (2013), Klumpp et al. (2019)).

4.1.2 Outline of the chapter

The rest of the article is organized as follows: Section 4.2 introduces the formal base model (the single-agent decision problem). Section 4.3 presents the optimal strategies and expected payoffs for the base model. Section 4.4 discusses the extensions to the base model along with their corresponding results. Section 4.4.4 presents a detailed comparison of EWR and optimal strategies in each of the extensions with respect to the base model. A conclusion follows.

4.2 Base model

Recall that we are interested in the problem of a coach in a competitive basketball league, who is required to make decisions about the overall effort level that the team exerts in each game, across the whole season. Let T denote the number of games that the *team* is scheduled to play in the season. The sequence of opponents that the coach faces over the entire season is collectively modelled as the *field*. At the start of each *game*, effort-reserves are realized for both the *team* and the *field*, denoted respectively by e^P and e^F. Score-difference at the end of the third quarter (henceforth *advantage*, denoted by A) is probabilistically determined as a function of effort exerted by the teams throughout the first three quarters combined.

What happens "late" in the game is salient in our model, and we assume that an exogenously given constant fraction α of the effort reserve is used up by the *team* throughout the first three quarters combined. In a similar way, the field is assumed to expend β fraction of its energy reserve throughout the first three quarters combined. Moreover, the *field* does not behave strategically in Q4 and therefore uses the remaining $(1-\beta)$ fraction in Q4. We relax this assumption in Section 4.4.1.

At the beginning of the fourth quarter (Q4), the coach has to decide between how much more effort to have his players exert in the current game; and how much to save for future games; as a function of the *advantage* at the beginning of Q4 and the effort reserve of the *field*. Effort reserves are replenished (stochastically) before the start of the next game. If there are no savings made in the current game, effort-reserve in the next game is continuously distributed with density g. If a saving of r is made in the current game, the effort reserve in the next game is drawn from a distribution with density $\mu(.|r)$, with more savings leading to the effort reserve in the next game being drawn from a better distribution, in a first order stochastic dominance sense. The coach's goal is to maximize the sum of stage rewards (winning probabilities in every game) across the whole season.

We model decisions as a discrete-time finite-horizon Markov Decision Process consisting of the following components. T denotes the number of *games* to be played by the *team* in the season and is indexed by t. Throughout the article, we will use the terms *game(s)* and *period(s)* to mean the same thing. Also, note that *period* is often used to refer to a quarter in basketball, but we always use it in the context of dynamic programming. The *state* variable in every game t is a triple $s_t = (e_t^F, e_t^P, A_t)$ where effort reserves e_t^F, $e_t^P \in [0,1]$; *advantage* at the start of Q4, $A_t \in \{-3p, ..., 0, ..., 3p\}$ where, $p > 0$ denotes the maximum possible point (score) difference between teams in a quarter. The assumption of maximum score difference p is made for simplicity — while there is no such score difference, in practice there would be — it is rare for an NBA team to outscore the other by more than 20 more points in a quarter. Positive value of A_t means the *team* is ahead at the start of Q4. The set of all possible states is denoted by $S = [0,1] \times [0,1] \times \{-3p, \cdots, 3p\}$. The effort spent by the *team* throughout the first three quarters combined is αe^P, where α is an exogenous parameter. We relax this assumption in Section 4.4.2 and let the *team* act strategically also during the first three quarters. The *control* variable $x_t \leq (1-\alpha)e_t^P$ denotes the effort exerted by the *team* in the fourth quarter (Q4) in game t. Finally, $r_t(x_t) = (1-\alpha)e_t^P - x_t$ denotes the effort saved in game t. The season starts with energy reserves e_1^P and e_1^F, and an *advantage* A_1 (distributed on $\{-3p, ..., 0, ..., 3p\}$ with mass function $\eta(\,.\mid \beta e_1^F,\ \alpha e_1^P))$ which is realized before the coach takes an action. We impose assumptions on the mass function which are discussed below (see Assumption 4.2.3).

Stage Reward

The team wins game t if the cumulative score difference after Q4 is positive. The score difference in Q4 is stochastically determined as a function of effort choices by the *team* and the *field* in Q4. Formally, the stage reward for the *team* is simply the probability of winning the game as a function of the *state* and the *control*:

Notation	Definition
T	The total number of games in the season.
e^P	Effort reserve of the team.
e^F	Effort reserve of the field.
A	Advantage (score-difference) at the start of $Q4$. Positive value indicates that the team is ahead.
p	A feasible maximum possible point (score) difference between the teams in a quarter.
s	A triple denoting the state, $s = (e^F, e^P, A)$.
S	State space, $S = [0,1] \times [0,1] \times \{-3p, \cdots, 3p\}$.
α	Fraction of effort reserve spent by the team throughout the first three quarters combined.
β	Fraction of effort reserve spent by the field throughout the first three quarters combined.
$\eta(\cdot \mid \beta e^F, \alpha e^P)$	Mass with which advantage (A) at the start of $Q4$ distributes, if effort reserves at the start of the game for the field and the team are e^F and e^P respectively.
x	Effort exerted by the team in $Q4$.
r	Effort saved by the team in $Q4$.
$g(\cdot)$	Density with which effort reserve in the next game distributes, if no saving is made in current game.
$\mu(\cdot \mid r)$	Density with which effort reserve in the next game distributes, if a saving of r is made in current game.
$\rho(\cdot \mid (1-\beta)e^F, x)$	Mass with which score-difference (X_4) in $Q4$ is distributed.

Table 4.1: Summary of notations.

$$R(s_t, x_t) = \begin{cases} P[\,\mathbf{X}_4 + A_t > 0\,] + \frac{1}{2}P[\,\mathbf{X}_4 + A_t = 0\,] & , A_t \in \{-p, -p+1, ..., p\} \\ 1 & , A_t \in \{p+1, p+2, ..., 3p\} \\ 0 & , A_t \in \{-3p, -3p+1, ..., -(p+1)\} \end{cases} \tag{4.1}$$

where, $\mathbf{X}_4$ denotes the score difference in Q4 and is discretely distributed on $\{-p, ..., 0, ..., p\}$ having probability mass function $\rho(\,.\mid x_t, (1-\beta)e_t^F)$.

Note, the way the *field* uses its remaining energy in Q4 is embedded in ρ. Also note that, if the cumulative scores are tied at the end of $Q4$, there is a probability of half for the *team* to win. The tie-breaker can be a result of, for instance, the game going into overtime, with the teams evenly matched. We summarize the notations described thus far in Table 4.1.

The state of the system at period $t+1$ depends on action x_t and a random shock unknown at period t as under:

- e^F_{t+1} is continuously distributed on [0,1] with density g
- e^P_{t+1} is continuously distributed on [0,1] with density $\mu(\,.\,|\,r_t(x_t))$
- A_{t+1} is distributed on $\{-3p,...,0,...,3p\}$ with mass $\eta(\,.\,|\;\beta e^F_{t+1},\;\alpha e^P_{t+1})$

Note that, the field's action is embedded in η. Recall that α and β are exogenously given. Next, we make the following assumptions on ρ, μ and η:

Assumption 4.2.1. *For $r > r' \geq 0$, $\mu(\,.\,|\,r)$ first-order stochastically dominates (F.O.S.D) $\mu(\,.\,|\,r')$.*

Assumption 4.2.2. *For $x > \tilde{x}$, $\rho(\,.\,|\,x\,,\,(1-\beta)e^F)$ F.O.S.D $\rho(\,.\,|\,\tilde{x}\,,\,(1-\beta)e^F)$ for any fixed $e^F \in [0,1]$. For $e^F < \tilde{e}^F$, $\rho(\,.\,|\,x\,,\,(1-\beta)e^F)$ F.O.S.D $\rho(\,.\,|\,x\,,\,(1-\beta)\tilde{e}^F)$ when x is held fixed.*

Assumption 4.2.3. *For $e^P > \tilde{e}^P$, $\eta(\,.\,|\,\alpha e^P,\,\beta e^F)$ F.O.S.D $\eta(\,.\,|\,\alpha\tilde{e}^P,\,\beta e^F)$ for any fixed $e^F \in [0,1]$. For $e^F < \tilde{e}^F$, $\eta(\,.\,|\,\alpha e^P,\,\beta e^F)$ F.O.S.D $\eta(\,.\,|\,\alpha e^P,\,\beta\tilde{e}^F)$ for any fixed $e^P \in [0,1]$.*

The first assumption captures the idea that higher energy saved in the current game leads to the effort reserve of the *team* in the next game being drawn from a better distribution, in a first order stochastic dominance sense. The second assumption guarantees that, when the effort of the *field* (resp., *team*) is held fixed, greater effort exertion by the *team* (resp., *field*) leads to better a score-difference (in $Q4$) in its favour, again in a first order stochastic dominance sense. The third assumption captures the same idea as the second, except that the score-difference in question is for the first three quarters combined.

Let $V_t(s)$ denote the value function capturing the maximum attainable sum of current and expected future rewards given that the system is in state $s = (e^F, e^P, A)$

in game t. Recall that, $S = [0,1] \times [0,1] \times \{-3p, \cdots, 3p\}$ denotes the state space. Then, the principle of optimality implies that the value functions $V_t : S \to \mathbb{R}$ must satisfy the *Bellman* equations :

$$V_t(s) = \max_x \left[R(s,x) + \int\int \{ \sum_a V_{t+1}(e^P, e^F, a) \eta(\, a \,|\, \alpha e^P, \beta e^F \,) \} \, g(de^F) \mu(de^P | r(x)) \right]$$

for all $s \in S$ and $t = 1, 2, ..., T$.

The tournament ends after *game* T, therefore we have:

$$V_{T+1}(s) = 0 \quad \forall \; s \in S$$

The next proposition states that indeed higher effort levels are more valuable for the *team* for every fixed effort level of the *field*, *advantage*, and time in the season. Thus, the model is consistent with our most basic intuition.

Proposition 4.2.1. *Fix any* α, $\beta \in [0,1]$. *Then,* $V_t(e^P, e^F, a)$ *is increasing in* e^P *for all* t, e^F, *and* a.

Proposition 4.2.1 is proved in Appendix C.

Next we prove a proposition for a more general Markov Decision Process than the one used in our model.

Proposition 4.2.2. *Consider a Markov Decision Process with* T *periods, state space* $\bar{S}$ *and a stage reward function* f *such that* $|f| \leq 1$.
If for initial states $s', s'' \in \bar{S}$, *and for any strategy* σ_1, *there exists strategy* σ_2 *such that in the processes beginning with* $s'_0 = s'$ *and* $s''_0 = s''$ *there is* N, $\varepsilon > 0$ *such that* $P_{\sigma_1,\sigma_2}\left[\exists \, k \leq N \;|\; V(s''_k) \geq V(s'_k) \right] > 1 - \varepsilon$ *then,*

$$\frac{V(s''_0)}{T} \geq \frac{V(s'_0)}{T} \; - \; error(\varepsilon, T, N)$$

where, *error*(ε *, T, N*) $= \left[T+3N+4\right] \cdot \frac{\varepsilon}{T} + \frac{4N}{T}$

Proposition 4.2.2 is proved in Appendix C. The intuition is as follow: no matter which strategy σ_1, the decision-maker can follow σ_2 to drive the game to a state which is very likely close to the value of states the strategy σ_1 reaches.

Corollary 4.2.1. *The error term above,* error(ε *, T, N*) $\to 0$ *as* $\varepsilon \to 0$, $T \to \infty$ *Therefore, for any two states* s'_0 *and* s''_0*,*

$$\left| \frac{V(s'_0)}{T} - \frac{V(s''_0)}{T} \right| \to 0$$

Next, we show that the conditions of Proposition 4.2.2 are satisfied in our setup. In our model, the stage reward is just the probability of winning the current game and therefore cannot exceed one. Recall that the state variable s_k in game k is a triple (e_k^F, e_k^P, A_k). Also note that, the choice of strategy only affects the evolution of the process $\{e_k^P\}$ and consequently the process $\{A_k\}$, but not the process $\{e_k^F\}$. To simplify notation, for the remainder of this section, we drop the superscript P and denote the process $\{e_k^P\}$ more simply as $\{e_k\}$. In the base model, since the *team* necessarily uses up fraction α of its energy reserves in the first three quarters combined, the maximum energy savings possible at $Q4$ in any game is $1-\alpha$, i.e., $r_k \leq 1-\alpha \ \ \forall k$. Therefore, in any game k, and for strategies σ_1 and σ_2, even if one strategy saves nothing while the other saves as much as possible, denoting P instead of P_{σ_1,σ_2} we have for any s'_{k-1}, s''_{k-1},

$$P[e'_k > e''_k \mid s'_{k-1}, s''_{k-1}] \ \leq \ \int_0^1 \int_0^{e'} \mu(de''|r=0)\,\mu(de'|r=1-\alpha) \ =: \ q \ < \ 1 \quad (4.2)$$

Let us define strategy σ_2 to be such that the coach saves all remaining energy at $Q4$. Now, given s'_{k-1}, s''_{k-1}, if team energy level in process $''$ is higher as one enters $Q4$ of game k, then to begin game $k+1$, the following holds:

* σ_2 has more energy than σ_1 with probability $\geq \frac{1}{2}$, by Assumption 4.2.1.

* with probability $\frac{1}{2}$, σ_2 faces a lower energy rival than σ_1, as field's energy in each process is chosen independently according to density g.

* conditional on the above, with probability $\geq \frac{1}{2}$, $Q4$ of game $k+1$ starts with σ_2's score-difference better than that of σ_1, by Assumption 4.2.3.

From the above three points, we have:

$$P\big[V(s''_{k+1}) \geq V(s'_{k+1}) \mid e''_k \geq e'_k,\, s'_{k-1}, s''_{k-1}\big] \;\geq\; \frac{1}{8} \tag{4.3}$$

Using (4.2) and the above we get,

$$\begin{aligned} P\big[V(s''_{k+1}) \geq V(s'_{k+1}) \mid s'_{k-1}, s''_{k-1}\big] \geq\;& P\big[e''_k \geq e'_k,\, V(s''_{k+1}) \geq V(s'_{k+1}) \mid s'_{k-1}, s''_{k-1}\big] \\ \geq\;& \frac{1}{8}(1-q) \\ =:\;& \hat{q} > 0 \end{aligned} \tag{4.4}$$

Therefore we have,

$$\begin{aligned} P\Big[\exists\, k \leq N \;\mid\; V(s''_{k+1}) \geq V(s'_{k+1})\Big] &= 1 - P\Big[\forall\, k \leq N \;\mid\; V(s'_{k+1}) > V(s''_{k+1})\Big] \\ &\geq 1-(1-\hat{q})^N \quad \text{(follows inductively using (4.4))} \end{aligned}$$

Thus, since $\hat{q} > 0$, for any $\varepsilon > 0$, choosing N large enough such that $(1-\hat{q})^N < \varepsilon$ we have, $P\Big[\exists\, k \leq N \;\mid\; V(s''_k) \geq V(s'_k)\Big] > 1-\varepsilon$.

Thus, the above-stated Corollary 4.2.1 highlights the insensitivity of initial conditions in a long season. Towards the beginning of a long season, for any two states, the difference between the average value in remaining games (i.e. the expected win rate in games remaining), becomes negligible.

4.3 Analysis and Results

We solve the optimization problem computationally. We make specific choices for the previously described probability distributions governing the process. These, along with the particular choice of all parameters of the model are presented below:

Distributions and Parameters

- Total number of games, T = 80
- Effort of Field, $e^F \sim$ Uniform $(0,1)$
- Effort of Team, $e^P \sim$ Uniform $(r,1)$ when saving in previous game is r
- Maximum absolute score difference in each quarter, $p = 10$.
- Fraction of total effort that the *team* uses in the first three quarters, $\alpha = 0.7$
- *Field* divides available effort reserve equally across all four quarters in every game. Therefore, $\beta = 0.75$.
- Advantage (A, with support $\{-30,-29,\cdots,0,\cdots,29,30\}$) to *team* at the start of $Q4$,

  $A \sim$ Binomial $\left(60, \frac{\alpha e^P}{\alpha e^P + \beta e^F}\right) - 30$
- Score difference in the fourth quarter (X_4, with support $\{-10,\cdots,0,\cdots,10\}$), given the effort exerted by the *team* in Q4 is x,

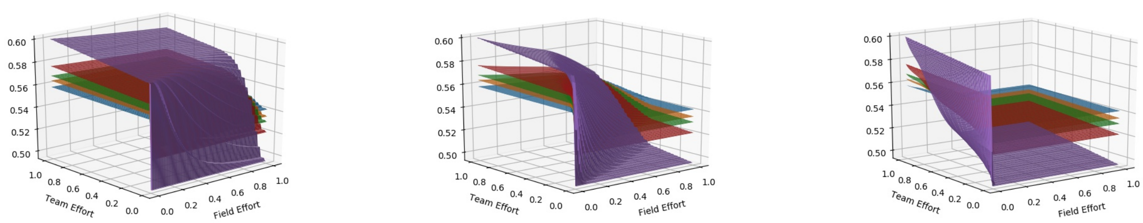
  $X_4 \sim$ Binomial$\left(20, \frac{x}{x+(1-\beta)e^F}\right) - 10$

The maximization problem in (Section 4.2) is solved by backward recursion. In order to facilitate computation, we discretize the state space. The domain of effort reserves $[0,1]$ for both the *team* and the *field* is discretized to 100 equispaced points. The third state component *advantage* is already discrete to begin with, ranging between $(3\times 2p)+1 = 61$ distinct values. Together this makes up the required $100\times 100\times 61$ grid points.

The recursion algorithm is structured as a series of three nested loops. The outer loop involves the backward recursion over *games*; the middle loop involves visits to each state (grid point); and the inner loop involves visits to each possible action (effort saved in the current game) to eventually evaluate the best action.

4.3.1 Expected Win Rate

For every state $s \in S$, the Expected Win Rate (EWR) in games remaining at period t is given by the expression $\frac{V_t(s)}{T-t+1}$. Figure 4.1 below shows the EWR during different stages of the season, under three scenarios of *advantage* ("ahead", "even", or "behind") that the team might face at the start of Q4.



(a) Ahead (b) Even (c) Behind

Figure 4.1: Expected Win Rate at different stages of the season. Different shades denote different game (t), with the following colour codes: t = 10 , 40 , 50 , 60 , 70

Remarks:

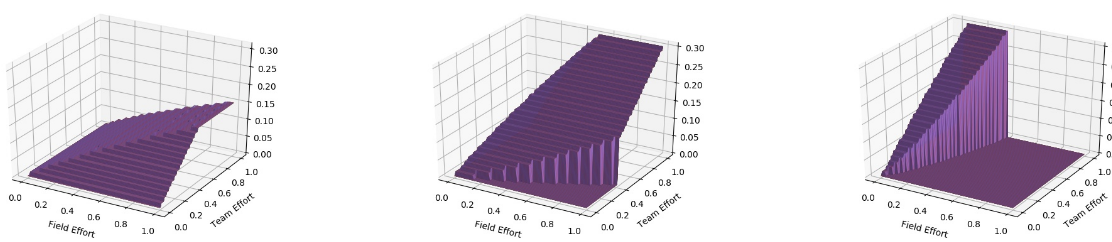
- The range of EWR shrinks as we go earlier in the season. This is consistent with Proposition 4.2.2, which highlights the insensitivity of initial conditions in a long season.

- When the team is well *ahead* or *behind* at the start of $Q4$, EWR does not change significantly with respect to the strength difference between the teams; except in the case when, either the *team* or the *field* starts with a very low reserve (close to zero). It is rather the time in the season that matters more. The value of being *ahead* (resp., *behind*) at the start of $Q4$, increases (resp., decreases) when there are fewer games remaining (see, Figure 4.1a & 4.1c).

- If scores are *even* at the start of $Q4$, the strength difference between the teams plays a role in determining EWR. Starting the game as the relatively stronger team is more valuable when there are fewer games remaining. On the other hand, the value of starting as the relatively weaker team gets increasingly worse as one proceeds through the season (Figure 4.1b).

4.3.2 Optimal Strategy

Figure 4.2 below shows the optimal strategy for the *team* as a function of its own effort level and the *field*'s effort level, at different stages of the season, for three *advantage* scenarios that the team might face at the start of $Q4$. Note that the range of the $z-axis$ is up to 0.30. This is because the team has spent αe^P ($\alpha = 0.70$) in the first three quarters.



(a) Ahead (b) Even (c) Behind

Figure 4.2: Optimal Strategy at different stages of the season. Different shades denote different game (t), with the following colour codes: t = 10 , 40 , 50 , 60 , 70

Remarks:

- Firstly, one can observe from Figure 4.2 that the optimal strategy is highly insensitive to the stage in the season.
- If the *advantage* at the start of $Q4$ is well in favour of the *team*, but the *team* starts with a very low initial energy reserve, it is optimal to use up the remaining energy against a relatively stronger opponent. However, starting the game with moderate energy reserves, good advantage always calls for saving energy in $Q4$ (Figure 4.2a).
- When the relative strength is strongly in favour of the *team*, and the scores are level at the start of $Q4$, we can see from Figure 4.2b that it is indeed optimal to

save energy for the future. If the *team* starts the game very low on energy, it is again optimal to save when scores are level after $Q3$.

- If the score-difference is largely in favour of the *field* after $Q3$, unless it is the case that the opponent is very weak compared to the *team*, Figure 4.2c shows that it is optimal to save energy in $Q4$ (there is a sharp drop to zero).

4.3.3 End Season

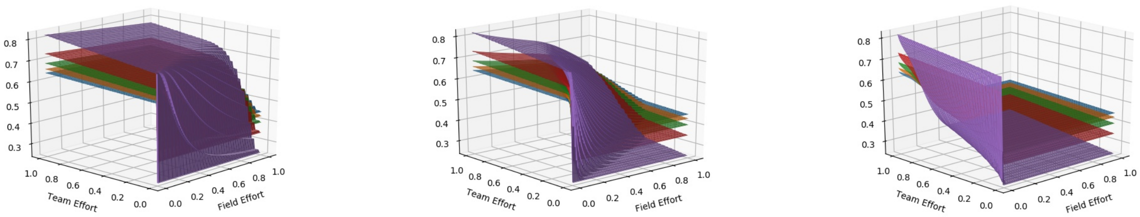
In this subsection, we focus attention to the last five games in the season in which, future games must be taken into consideration ($t = 75, 76, 77, 78, 79$). Recall that the game ends when $t = T = 80$, and thus the *team* does not need to save for future games and uses up all the remaining energy $(1-\alpha)e^P_{80}$ in $Q4$ of game 80. By last five games, we will henceforth refer to the following games (t) — $75, 76, 77, 78$ and 79.

Expected Win Rate

- There is a very large variation in the values of EWR across states when compared to games earlier in the season (Figure 4.3).

- If the *team* is well *ahead* or *behind* at the start of $Q4$, the strength difference between the teams does not lead to a significant change in EWR; except in the case when, either the *team* or the *field* starts with a very low reserve (close to zero). The value of being *ahead* (resp., *behind*) at the start of $Q4$ is quite high (resp., low); and increases (resp., decreases) further as one proceeds to the very end of the season (see, Figure 4.3a & 4.3c).

- If scores are *even* at the start of $Q4$, the strength difference between the teams plays a key role in determining EWR. Starting the game as the relatively stronger team is more valuable when there are fewer games remaining. On the other hand, the value of starting as the relatively weaker team is increasingly worse, as one heads to the very end of the season (Figure 4.3b).

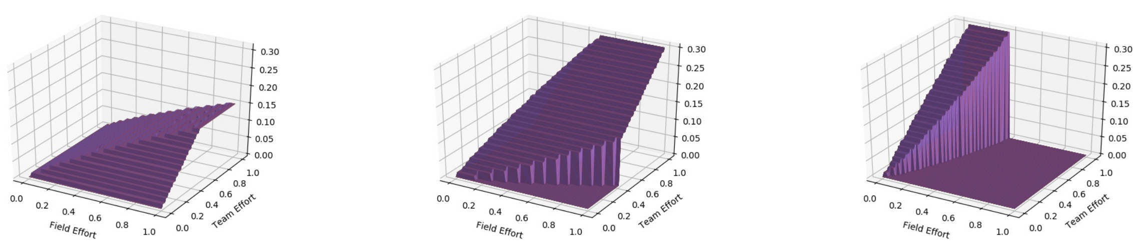


(a) Ahead (b) Even (c) Behind

Figure 4.3: Expected Win Rate in the last five games (t). Colour codes: t = 75 , 76 , 77 , 78 , 79

<u>*Optimal Strategy*</u>

- The optimal strategy is quite insensitive to which game we consider even towards the very end of the season (Figure 4.4).
- Saving energy in $Q4$ is indeed optimal in a variety of situations, similar to what we observed for games earlier in the season as shown in Figure 4.2.



(a) Ahead (b) Even (c) Behind

Figure 4.4: Optimal Strategy at the end of the season. Colour codes: t = 75 , 76 , 77 , 78 , 79

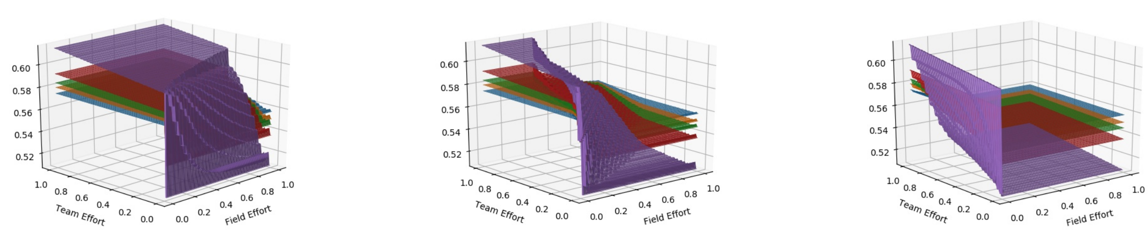
4.4 Extensions

4.4.1 Extension I : Field plays strategically

In this section, we analyze the case where the *field* plays the optimal strategy evaluated in the base model (see, Section 4.3.2). This optimal strategy used by the *field* in every game, is appropriately embedded in η and ρ, which dictates the outcome of random variables A (advantage) and X_4 (score difference in the last quarter). Given this use of *field*'s strategy, we re-evaluate the *team*'s best response. The maximization problem in (Section 4.2) is again solved by backward induction over a discretized state space as before. Next, we discuss its implications on the *Expected Win Rate* and the *Optimal Strategies*.

Expected Win Rate

- The range of EWR shrinks as we go earlier in the season, highlighting the insensitivity of initial conditions in a long season (Figure 4.5).
- If the $team$ is well *ahead* or *behind* at the start of $Q4$, the strength difference between the teams does not lead to a drastic change in EWR; except in the case when, either the $team$ or the *field* starts with a very low reserve (close to zero). It is rather the time in the season that matters more. The value of being *ahead* (resp., *behind*) at the start of $Q4$, increases (resp., decreases) when there are fewer games remaining (see, Figure 4.5a & 4.5c).
- If scores are *even* at the start of $Q4$, the strength difference between the teams plays a role in determining EWR. Starting the game as the relatively stronger team is more valuable when there are fewer games remaining. On the other hand, the value of starting as the relatively weaker team gets increasingly worse as one proceeds through the season (Figure 4.5b).



(a) Ahead (b) Even (c) Behind

Figure 4.5: Expected Win Rate at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 10 , 40 , 50 , 60 , 70

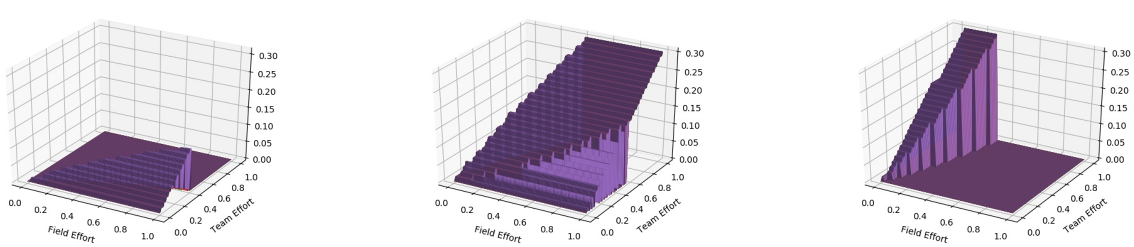
Optimal Strategy

- Firstly, one can observe from Figure 4.6 that the optimal strategy is highly insensitive to the stage in the season.

- When the *advantage* at the start of $Q4$ is well in favour of the *team*, but the *team* starts with a very low initial energy reserve, it optimal to use up the remaining energy against a relatively very strong opponent. However, for all other starting energy scenarios, good advantage always calls for saving energy in $Q4$ (there is a sharp drop to zero, see Figure 4.6a).

- If the scores are level at the start of $Q4$ and the relative strength is strongly in favour of the *team* (i.e., *field* starts with energy reserve close to zero); we see from Figure 4.6b that it is indeed then optimal to save energy for future games. If the *team* starts the game very low on energy, there are occasional spikes of energy utilization in $Q4$, even when the *team* faces a relatively strong opponent (see, Figure 4.6b).

- If the score-difference is largely in favour of the opponent after $Q3$, unless the opponent starts the game very weak (energy close to zero) compared to the *team*, it is always optimal to save all remaining energy in $Q4$ (there is a sharp drop to zero, see Figure 4.6c).



(a) Ahead (b) Even (c) Behind

Figure 4.6: Optimal Strategy at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 10 , 40 , 50 , 60 , 70

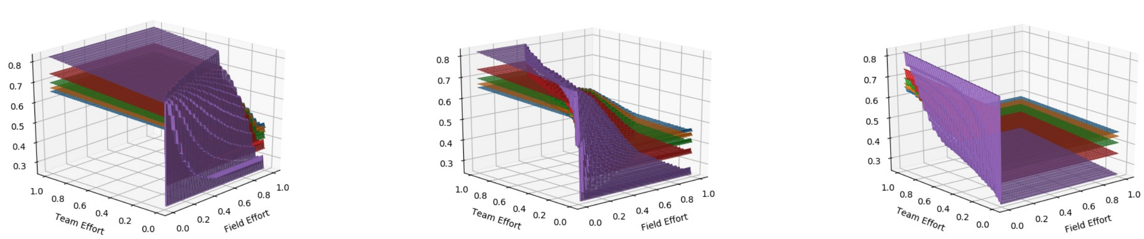
End Season

Expected Win Rate

- There is a very large variation in the values of EWR when compared to games earlier in the season (Figure 4.7).

- If the *team* is well *ahead* or well *behind* at the start of $Q4$, the strength difference between the teams does not affect EWR; except when, either the *team* or the *field* starts with a very low reserve (close to zero). The value of being *ahead* (resp., *behind*) at the start of $Q4$ is quite high (resp., low); and increases (resp., decreases) further as one proceeds to the very end of the season (see, Figure 4.7a & 4.7c).

- If scores are *even* at the start of $Q4$, the strength difference between the teams plays a key role in determining EWR. Starting the game as the relatively stronger team is more valuable when there are fewer games remaining. However, the value of starting as the relatively weaker team gets increasingly worse as we head to the very end of the season (Figure 4.7b).

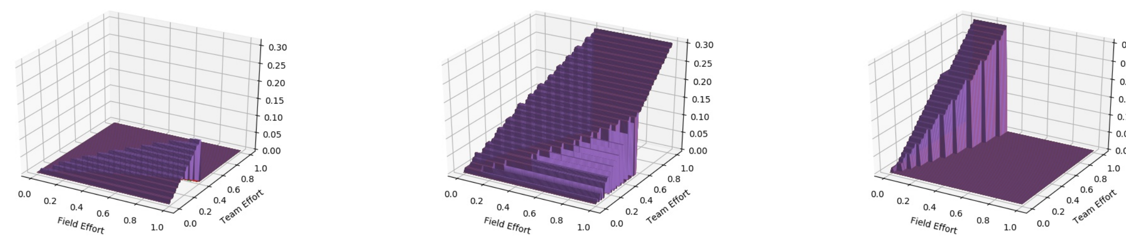


(a) Ahead (b) Even (c) Behind

Figure 4.7: Expected Win Rate at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 75 , 76 , 77 , 78 , 79

Optimal Strategy

- The optimal strategy is quite insensitive to which game we consider even towards the very end of the season (Figure 4.8).
- Saving energy in $Q4$ is indeed optimal in a variety of situations, similar to what we observed for games earlier in the season as shown in Figure 4.6.

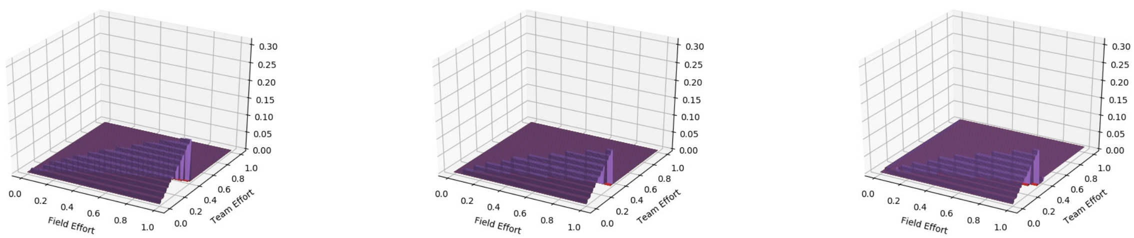


(a) Ahead (b) Even (c) Behind

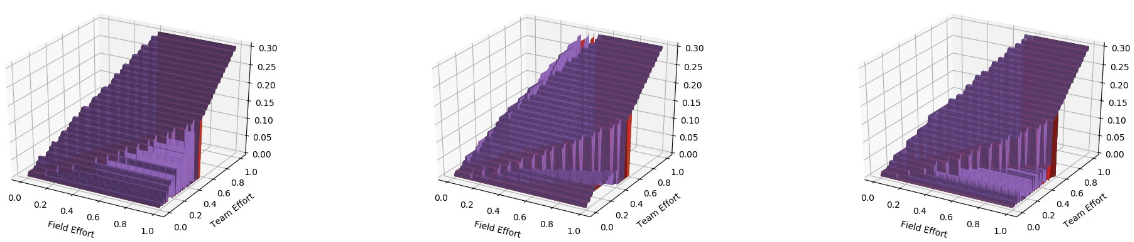
Figure 4.8: Optimal Strategy at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 75 , 76 , 77 , 78 , 79

Iterated evaluation of Optimal Strategies

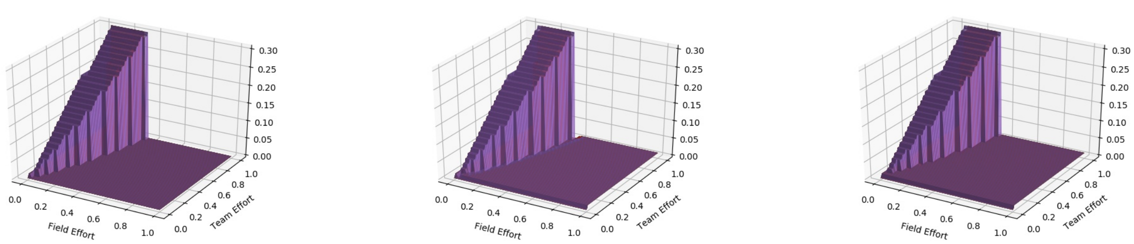
In this subsection, we show that the optimal strategies when the field plays strategically (as presented in the Section 4.4.1); if played by both the *team* and the *field*, in fact, constitutes an approximate equilibrium of the game. In order to show this, we iteratively solve the maximization problem of the *team* by letting the *field* play the optimal strategy calculated in the previous iteration. That is to say, at every iteration, the optimal strategy in the previous iteration is plugged into the maximization problem as the field's strategy, and a new optimal strategy is then evaluated. We show that there is no significant change in the optimal strategies across three iterations (*Iteration 2*, *Iteration 3* & *Iteration 4*), suggesting that we have indeed reached an approximate equilibrium. Figure 4.9 shows the comparison of optimal strategies across the different iterations, for three *advantage* scenarios: *ahead*, *even*, *behind*. Note that, *Iteration 2* constitutes the simulation in which the optimal strategy presented previously in Figure 4.6 is employed by the field (for further details, see Section 4.4.1). This is presented alongside the optimal strategies from the next two consecutive iterations to make the comparison easier to visualize. Indeed, as can be seen in Figure 4.9, when the *team* is ahead or behind at the start of $Q4$ there is almost no variation in optimal action across the three iterations whatsoever. When scores are even, there is a slight discrepancy in the optimal actions, however, they seem to still be fairly close to each other overall.



(a) Advantage: Ahead



(b) Advantage: Even



(c) Advantage: Behind

Figure 4.9: Optimal strategies in three iterations. Going from left to right, optimal strategies in *Iteration 2* appears in the left most column, followed by *Iteration 3*, and finally *Iteration 4*.

4.4.2 Extension II : Strategic play by the team in all quarters

In this section, we relax the assumption on α, which recall was the fraction of energy used by the *team* during the first three quarters and was taken to be exogenously given. We instead allow for the coach to act strategically during the first three quarters (considered together) as well. Recall that T denotes the total number of *games* to be played by the *team* in the season and is indexed by t. Effort (energy) reserves of the *team* and the *field*, are denoted respectively by e^P and e^F. Score-difference (*advantage*) at the end of the third quarter (denoted by A) is probabilistically determined as a function of effort exerted by the teams throughout the first three quarters combined.

The *field* is assumed to split energy equally over all quarters. At the start of each game, energy reserves are realized. For each game t, there are now two decision making points — (1) at the beginning of the game, when the coach decides how much energy (denoted by x) to have his team spend in the first three quarters ($Q1$ to $Q3$ combined), as a function of e^F and e^P and; (2) at the start of $Q4$ after observing the *advantage*, the coach decides, out of the remaining energy $e^P - x$ in the game, how much to save (denoted by r) in consideration of future games versus how much to use in $Q4$ to win the current game. Let $Q^{<4}$ and Q^4 denote the above two decision making points (henceforth, rounds) in every game.

The above-described problem can again be modelled as a Markov Decision Process, but this time with $2 \times T$ rounds, *two* rounds (one each for $Q^{<4}$ and Q^4) for every *game* $t \in T$. The *state* variable at the start of round $Q^{<4}$ of game t is a pair $s_t^{Q^{<4}} = (e_t^F, e_t^P)$ where, energy reserves e_t^P, $e_t^F \in [0,1]$. The action variable in round $Q^{<4}$ of game t is $x_t \leq e_t^P$, which denotes the energy used in quarter $Q1$ to $Q3$. Given action x_t, the state (available energy and advantage) at the start of round Q^4 of game t then transitions to $s_t^{Q^4} = (\frac{e_t^F}{4}, e_t^P - x_t,\ A_t)$ where, advantage A_t is distributed on $\{-3p, ..., 0, ..., 3p\}$ [1] with mass function $\eta(\ .\mid\ 0.75e_t^F,\ x_t)$

[1] $p > 0$ denotes the maximum possible point (score) difference between teams in a quarter. Positive advantage denotes the team is ahead at the start of $Q4$.

[2]. Let $S^{Q^{<4}} = [0,1] \times [0,1]$ and $S^{Q^4} = [0,\frac{1}{4}] \times [0,1] \times \{-3p,\cdots,3p\}$ respectively denote, the set of all possible $Q^{<4}$ and Q^4 states. $S = S^{Q^{<4}} \times S^{Q^4}$ denotes the state space. The assumption on the mass function η remains the same as earlier (see Assumption 4.2.3). The action variable $r_t \leq e_t^P - x_t$ in round Q^4 of game t, denotes the energy saved in $Q4$ of game t. The reward function at $Q4$ of game t is as below:

Reward function

The team wins game t, if the cumulative score difference at the end of $Q4$ is positive. The score difference in $Q4$ is stochastically determined as a function of effort choices by the *team* and the *field* in $Q4$. Formally, the stage reward for the *team* is simply the probability of winning the game as a function of the *state* and the *control*

$$R\Big(\frac{e_t^F}{4}, e_t^P - x_t,\, A_t,\, r_t\Big) = \begin{cases} P[\,\mathbf{X}_4 + A_t > 0\,] + \frac{1}{2}P[\,\mathbf{X}_4 + A_t = 0\,] & ,\, A_t \in \{-p,-p+1,...,p\} \\ 1 & ,\, A_t \in \{p+1,p+2,...,3p\} \\ 0 & ,\, A_t \in \{-3p,-3p+1,...,-(p+1)\} \end{cases} \tag{4.5}$$

where, $\mathbf{X}_4$ denotes the score difference in Q4 and is discretely distributed on $\{-p,...,0,...,p\}$ having probability mass function $\rho(\,.\mid e_t^P - x_t - r_t,\, \frac{e_t^F}{4})$.

Note, the way the *field* uses its remaining energy in Q4 is embedded in ρ. Also note that, if the cumulative scores are tied at the end of $Q4$, there is a probability half of the *team* winning. The tie-breaker can be a result of, for instance, the game going into overtime, with the teams evenly matched.

Energy reserves in game $t+1$ depend on action r_t in $Q4$ and a random shock unknown at period t as below:

- e_{t+1}^F is continuously distributed on [0,1] with density g
- e_{t+1}^P is continuously distributed on [0,1] with density $\mu(\,.\mid r_t)$

[2]Recall that the field divides energy equally across all quarters, and hence 75% in first three quarters

Let $V_t^{Q^4}(s)$ denote the value function capturing the maximum attainable sum of current and expected future rewards given that the system is in state s in round Q^4 of game t.

Similarly, $V_t^{Q^{<4}}(s)$ denotes the value function in state s at the start of round $Q^{<4}$ of game t. Then the principle of optimality implies that the value functions must satisfy the *Bellman* equations :

$$V_t^{Q^4}\left(\frac{e_t^F}{4}, e_t^P - x_t, A_t\right) = \max_{r \leq e_t^P - x_t} \left[R\left(\frac{e_t^F}{4}, e_t^P - x_t - r, A_t\right) + E\left(V_{t+1}^{Q^{<4}}\left(e_{t+1}^F, e_{t+1}^P\right) \,\middle|\, r\right)\right]$$

$$V_t^{Q^{<4}}\left(e_t^F, e_t^P\right) = \max_{x \leq e_t^P} \sum_a V_t^{Q^4}\left(\frac{e_t^F}{4}, e_t^P - x,\ a\right) \eta\left(a \,\middle|\, x, \frac{3e_t^F}{4}\right)$$

for all e_t^F, e_t^P, A_t and $t = 1, 2, ..., T$

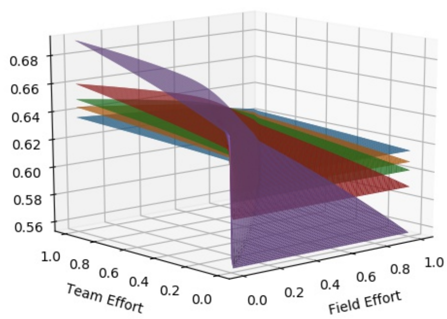
The tournament ends after *game* T, therefore we have:

$$V_{T+1}^q(s) = 0 \quad \forall\ s \in S^q\ \forall\ q \in \{Q^{<4}, Q^4\}$$

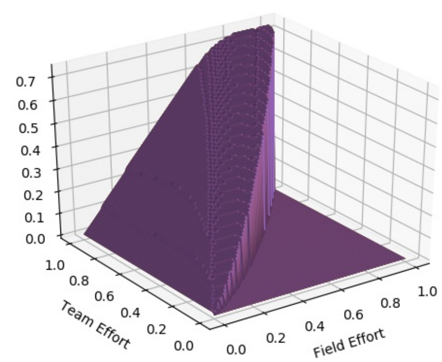
We once again solve the above optimization problem computationally by backward induction. We make specific choices for the previously described probability distributions governing the process. Assumptions about these distributions remain same as in earlier sections (for details see, **Distributions and Parameters** in Section 4.3). Next we present the results.

Expected Win Rate at the start of the game

- The range of EWR shrinks as we go earlier in the season (Figure 4.10a), similar to what was observed in the base model.
- Relative strengths between the teams play a role in determining EWR. Starting the game as the relatively stronger team is more valuable when there are fewer games remaining. However, the value of starting as the relatively weaker team gets increasingly worse as one proceeds through the season (Figure 4.10a).



(a) Expected Win Rate



(b) Energy Used in Q1-Q3

Figure 4.10: Expected Win Rate and Optimal Action in Q1-Q3 at different stages of the season. Different shades denote different game (t), with colour codes: $t =$ 10 , 20 , 40 , 60 , 70.

Optimal Action in Q1-Q3

- The optimal action (energy used in Q1-Q3) is highly insensitive to the stage in the season (Figure 4.10b).
- If the relative strength between the teams is tilted in favour of the coach, the optimal action weakly increases with *field*'s starting strength. Also it is interesting to note, the optimal energy used in the first three quarters never exceeds 0.75 no matter how strong the *team* starts the game (see, Figure 4.10b). This is not an obvious deduction at the beginning as one might think that it could perhaps

be optimal to exert higher effort during the first three quarters and gain a strong lead. However, we see that it is not the case.

- If the opponent is relatively stronger, it is optimal to save up entirely for the last quarter (there is a sharp drop to zero, see Figure 4.10b). Note that, an interpretation of "effort" in our model is from the point of view of resting or limiting minutes of star players, which is indeed something coaches consider frequently. So "putting zero effort" need not really mean the players on the field do not try, but rather that the coach decides to rest his stars.

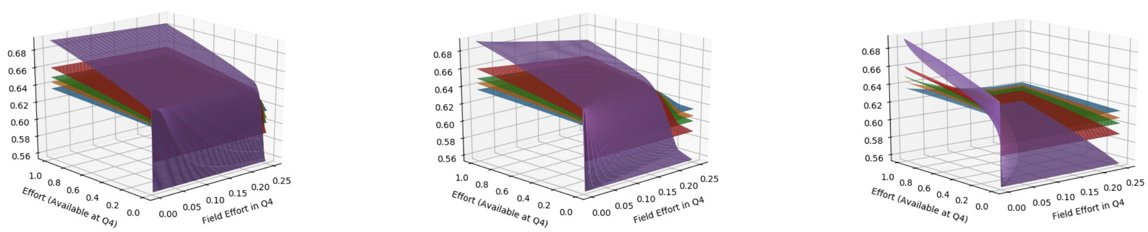
Expected Win Rate at the start of Q4

- Figure 4.11 show EWR at different *advantage* scenarios that the team might face at the start of $Q4$. Note the asymmetry in the possible remaining energy levels of the teams.

- If the *team* is well *ahead* at the start of $Q4$, the difference between the teams in terms of energy available at $Q4$ do not affect EWR; except when, the *team* starts $Q4$ with very low energy (close to zero). As long as the *team* does not start $Q4$ with energy close to zero, the value of being *ahead* at the start of $Q4$ increases as one proceeds through the season (see, Figure 4.11a).

- If the *team* is well *behind* at the start of $Q4$, the available-energy difference between the teams does not affect EWR; except if the *field* starts $Q4$ with a very low reserve (close to zero). If *field*'s energy level at $Q4$ is not very low, the EWR decreases as one proceeds through the season. If *field*'s energy level at $Q4$ is very low, EWR of starting $Q4$ very high in energy reserve increases as we progress into the season (see, Figure 4.11c).

- If scores are *even* at the start of $Q4$, available-energy difference between the teams plays a role in determining EWR. The EWR of starting $Q4$ as the relatively weaker team is increasingly worse, as one heads to the end of the season. The EWR of starting $Q4$ as the relatively stronger team increases as we proceed through the season. Once the *team*'s available energy at $Q4$ exceeds 0.25 (the highest energy level that the *field* can start $Q4$ with); the energy-difference does affect EWR any longer, only the time in the season matters (Figure 4.11b).



(a) Ahead (b) Even (c) Behind

Figure 4.11: Expected Win Rate at different stages of the season. Different shades denote different game (t), with the following colour codes: $t =$ 10 , 40 , 50 , 60 , 70

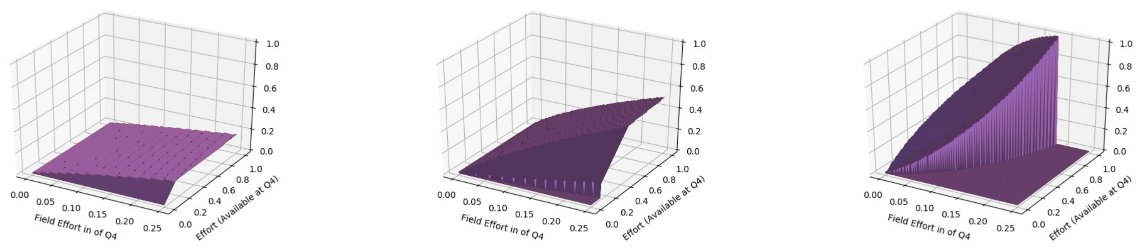
Optimal Action in Q4

- We can observe from Figure 4.12 that the optimal strategy is highly insensitive to the stage in the season.

- If the *advantage* at the start of $Q4$ is well in favour of the *team*, but the *team* starts with low energy in $Q4$, it optimal to use up the energy against a relatively strong opponent. However, for every fixed energy of the *field*, starting $Q4$ with energy approximately about 0.20 and higher, the optimal action levels out and saving energy becomes optimal (see, Figure 4.12a).

- If the *advantage* is *even* at the start of $Q4$, and the *team* starts $Q4$ low on energy, it is optimal to save all of it. If the *team* starts $Q4$ with moderate energy and higher, for every fixed energy of the *field*, the optimal action weakly increases with available energy at $Q4$ and eventually levels out beyond a threshold (depending on the *field*'s energy, see 4.12b).

- If the *advantage* is *behind* at the start of $Q4$, it is optimal to still use up energy in $Q4$ if the relative strength between the teams at $Q4$ is very very strongly in favour of the *team*. Otherwise, it is optimal to save all the available energy (there is a sharp drop to zero see, Figure 4.12c) .



(a) Ahead (b) Even (c) Behind

Figure 4.12: Optimal Energy usage in Q4 at different stages of the season. Colour codes: $t =$ 10 , 40 , 50 , 60 , 70

End Season

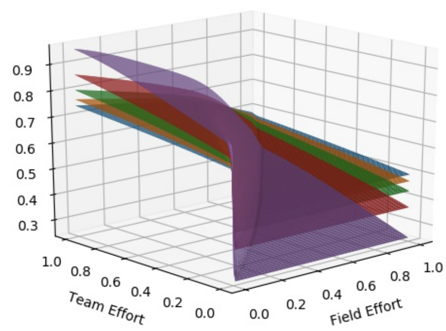
Next, we focus attention to the last five games in the season in which, future games must be taken into consideration ($t = 75, 76, 77, 78, 79$).

Expected Win Rate at the start of the game

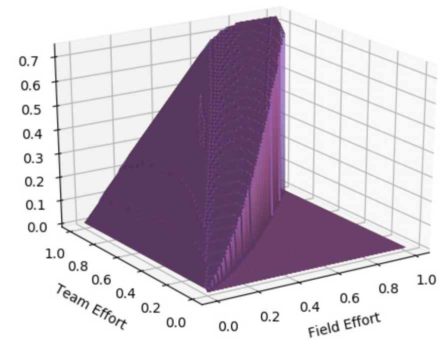
- Figure 4.13a shows EWR at the start of the game, for games towards the end of the season. The variation of EWR across states is much larger compared to games earlier in the season.
- Starting the game as the relatively stronger team is more valuable as we head to the very end the season. On the other hand, the EWR of starting as the relatively weaker team also gets increasingly worse towards the very end (Figure 4.13a).

Optimal Action in Q1-Q3

- The optimal action (energy used in Q1-Q3) remains very similar across games towards the end of the season (Figure 4.13b).
- So long as the relative strength between the teams is tilted in favour of the coach, the optimal action weakly increases with *field*'s starting strength, similar to earlier games in the season.



(a) Expected Win Rate



(b) Energy Used in Q1-Q3

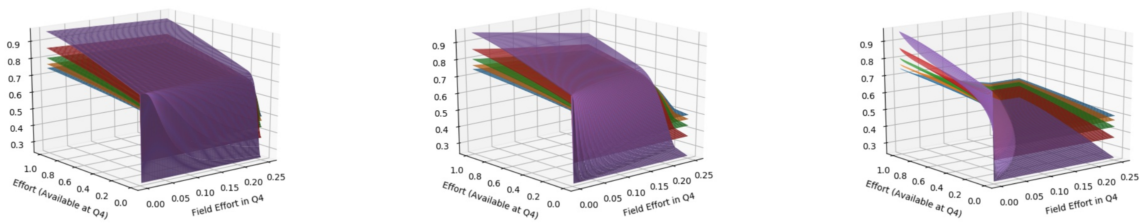
Figure 4.13: Expected Win Rate and Optimal Action in Q1-Q3 towards the end of the season. Different shades denote different game (t), with colour codes: $t =$ 75 , 76 , 77 , 78 , 79.

- If the opponent is relatively stronger, it is then optimal to save up entirely for the last quarter (there is a sharp drop to zero, see Figure 4.13b). Again note, interpreting "effort" from the point of view of resting or limiting minutes of star players, "putting zero effort" need not really mean the players on the field do not try, but rather that the coach decides to rest his stars.

Expected Win Rate at the start of Q4

- There is a larger variation in the values of EWR across states compared to games earlier in the season (Figure 4.14).

- The EWR varies with *advantage*, number of games remaining in the season, and the relative strength between the teams at the start of $Q4$; in the same way as in earlier games in the season. The effect (rise or fall of EWR) is however amplified compared to earlier games.

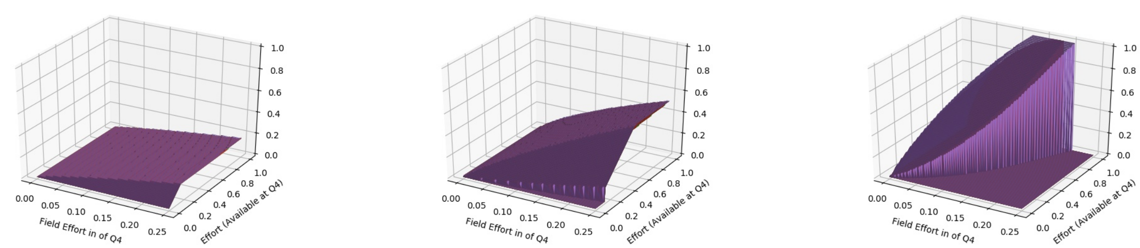


(a) Ahead (b) Even (c) Behind

Figure 4.14: Expected Win Rate in the last five games (t). Colour codes: t = 75 , 76 , 77 , 78 , 79

Optimal Action in Q4

- We can observe from Figure 4.15 that the optimal action is highly insensitive to which game we consider even towards the end of the season.
- Moreover, the optimal action resembles closely with the optimal action at $Q4$ for earlier games in the season.



(a) Ahead (b) Even (c) Behind

Figure 4.15: Optimal Strategy at the end of the season. Colour codes: t = 75 , 76 , 77 , 78 , 79

Optimal Strategies: Few specific scenarios

In this section, we dive deeper into the optimal strategies for three different initial energy levels of the *team*: strong, average and weak ($e^P = 0.8, 0.5$ and 0.2 respectively). We fix the game ($t = 40$) for the rest of this subsection.[3]

[3]Since the optimal action in $Q1 - Q3$ is highly insensitive to which game in the season the team is in (see Figure 4.10b); the optimal action in $Q4$ therefore also remains highly insensitive to the game number.

Team starts strong

Figure 4.16 shows the optimal energy usage by the team during $Q1$-$Q3$. The optimal action initially increases linearly with starting strength of the opponent, and then flattens out before sharply dropping to zero, against a very strong opponent ($e^F \approx 0.9$ and higher). Surprisingly, against a very strong opponent, strategically saving up big for the end is profitable even if the team starts the game quite strong ($e^P = 0.8$). Allowing the team to be strategic about how much effort to use in the first three quarters can be helpful (indeed, the EWR is higher than in the base model (see, Section 4.4.4)).

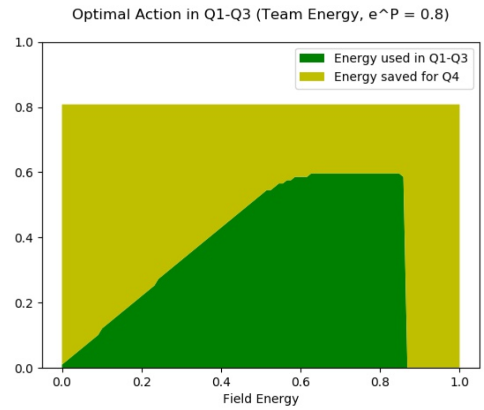
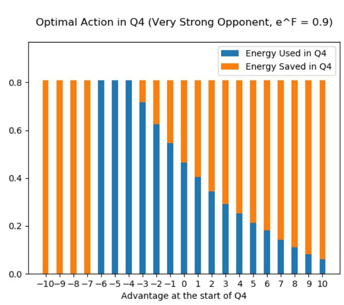
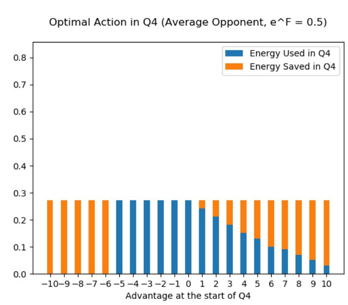


Figure 4.16: Optimal energy usage in the first three quarters combined when the team starts strong.

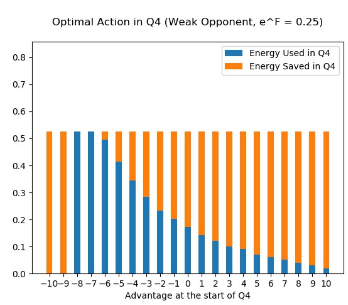
Figure 4.17 shows the optimal action in $Q4$, for different *advantage* scenarios that the *team* might face; against a *strong*, *average* and *weak* opponent. If the score-difference is not terribly bad, the coach can then try to overpower the strong opponent in the last quarter Figure (4.17a).



(a) Strong Opponent



(b) Average Opponent



(c) Weak Opponent

Figure 4.17: Optimal Action in $Q4$ when the team starts strong ($e^P = 0.8$).

Note, since the maximum possible score difference in a quarter was assumed to be 10, *advantage* below -10 and above 10 always calls for saving energy and is omitted from the presentation in Figure 4.17 and similar figures later. We observe that it is indeed optimal to save for future games in a range of scenarios.

Team starts average

Figure 4.18 shows the optimal energy usage by the team during $Q1$-$Q3$ when the team starts the game with average strength ($e^P = 0.5$). The optimal action initially increases linearly with strength of the opponent, and then flattens out before discontinuously dropping to zero, against a stronger opponent ($e^F \approx 0.6$ and above). Again we see, it is optimal for the coach when facing a team that has high energy to simply "not bother" and save its energy in the first three quarters. Depending on the score difference at the start of $Q4$, the coach then decides where it is worth investing or save for later games.

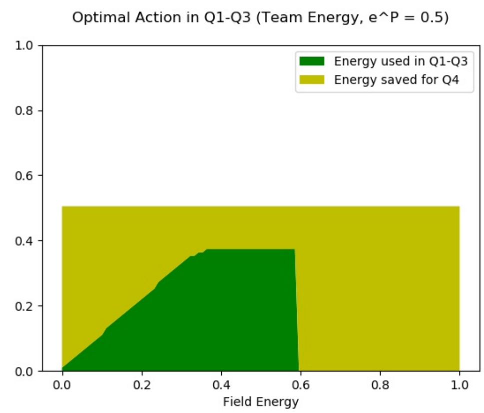
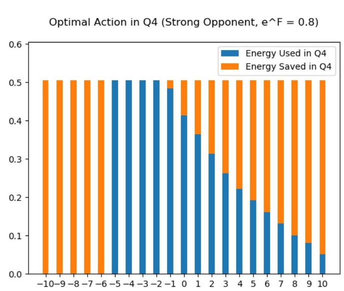
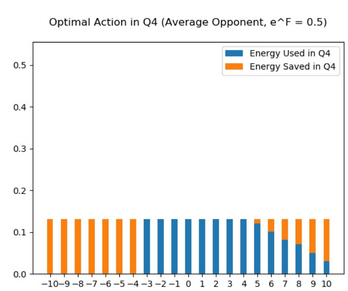


Figure 4.18: Optimal energy usage in the first three quarters combined when the team starts with average energy.

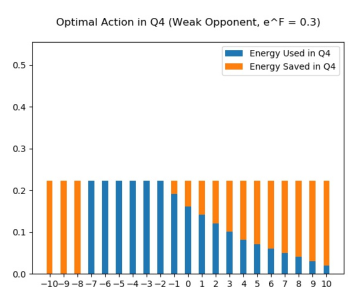
Figure 4.19 shows the optimal action in $Q4$, for different *advantage* scenarios that the *team* might face, at the start of the last quarter; against a *strong*, *average* and *weak* opponent.



(a) Strong Opponent



(b) Average Opponent



(c) Weak Opponent

Figure 4.19: Optimal Action in $Q4$ when the team starts average ($e^P = 0.5$).

Team starts weak

Figure 4.20 shows the optimal energy usage by the team during $Q1$-$Q3$ when the team starts the game with average strength ($e^P = 0.2$). Like in the previous case, the optimal action again initially increases linearly with strength of the opponent, and then flattens out before discontinuously dropping to zero, against a stronger opponent ($e^F \approx 0.3$ and above).

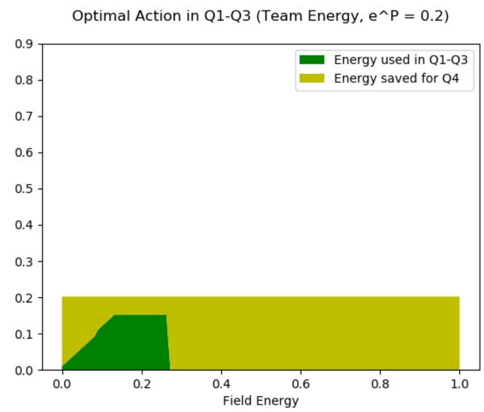
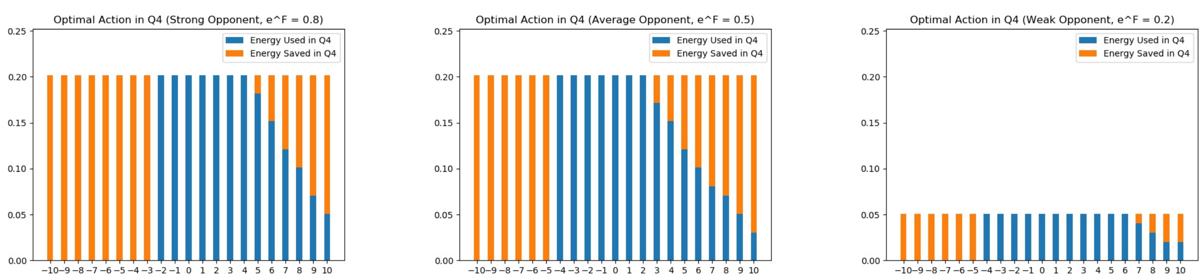


Figure 4.20: Optimal energy usage in the first three quarters combined when the team starts with average energy.

Figure 4.21 shows the optimal action in $Q4$, for different *advantage* scenarios that the *team* might face, at the start of the last quarter; against a *strong*, *average* and *weak* opponent.



(a) Strong Opponent (b) Average Opponent (c) Weak Opponent

Figure 4.21: Optimal Action in $Q4$ when the team starts weak ($e^P = 0.2$).

4.4.3 Extension III: Strong and weak opponents

Fixtures in competitions are usually known in advance. In this section therefore, we consider the case, where all opponents in the league are categorized into two types ("strong" and "weak") based on strength. The sequence of opponent strength is known to the *team* manager. We assume that the *team* faces roughly an equal number of strong and weak opponents in the season. The departure from the base model in Section 4.3 involves new assumptions about the domain of the *field*'s energy reserves. Energy reserves (e^F) for "strong" opponents are assumed to be distributed uniformly with support $[0.5, 1]$ while reserves for "weak" types also distributes uniformly, but with support $[0, 0.5]$. The assumptions about other distributions and parameters remain unchanged (for details see, **Distributions and Parameters** in Section 4.3). The expected value at $t+1$ in the maximization problem in (Section 4.2) is now evaluated over one of the truncated support, depending on the strength of the opponent at $t+1$ which is known to the *team*. We analyse EWR and optimal strategy of the *team*, during different stages of the season ($t = 5, 15, 40, 65, 75$), under three *advantage* scenarios as before. The sequence of opponent strength satisfying the conditions stated earlier is fixed. Table 4.2 presents the number of *strong* opponents remaining at different stages that we analyze. One can note that, at each stage, roughly an equal number of each type of opponent remain. Among the games presented, only at game (t) 15 there are fewer "strong" compared to "weak" opponents, that the *team* has yet to face.

Current Game	# Games Remaining	# Strong Opponents Remaining
5	75	38
15	65	31
40	40	21
65	15	9
75	5	3

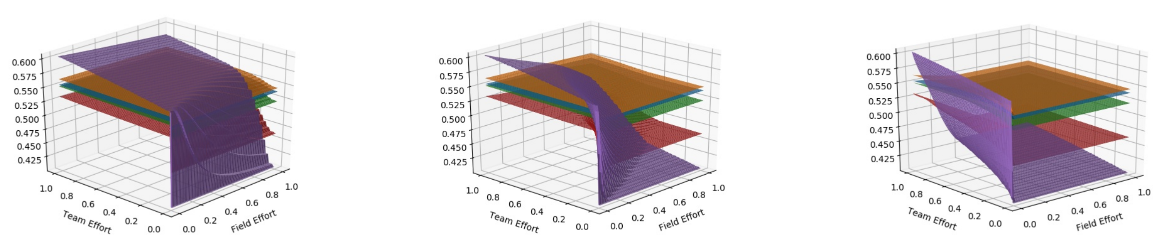
Table 4.2: Number of Strong Opponents remaining at different stages of the season.

Expected Win Rate

- During the first half of the season, EWR remains fairly constant across states, the exception being when either the *team* or the *field* starts with a very low reserve (close to zero); even in that case the change is not drastic (Figure 4.22).
- In the later half of the season, if the *advantage* at the start of $Q4$ is well *ahead* or *behind*, the strength difference between the teams does not affect EWR; except

when, either the *team* or the *field* starts with a very low reserve (close to zero). The combined effect of: (1) time in the season and (2) the number of strong opponents remaining is less obvious. (see, Figure 4.22a & 4.22c).

- If scores are *even* at the start of $Q4$, the strength difference between the teams plays a role in determining EWR. There is a large variation in the values of EWR depending on whether the *team* is relatively stronger or weaker than the field when there are fewer games remaining. Relative differences between EWR at different times of the season based on the current state is again less obvious (Figure 4.22b).



(a) Ahead (b) Even (c) Behind

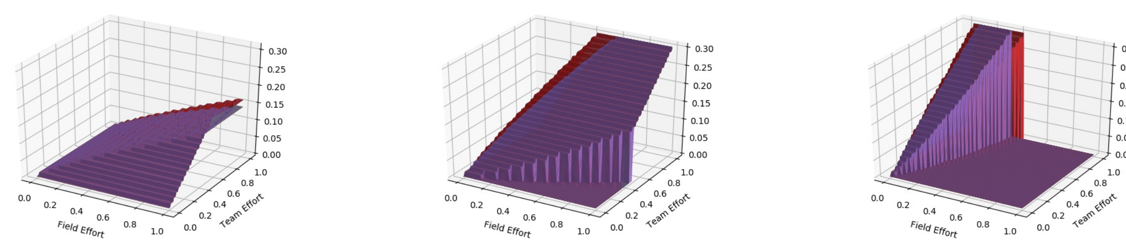
Figure 4.22: Expected Win Rate at different stages of the season. Different shades denote different game (t), with the following colour codes: t = 5 , 15 , 40 , 60 , 75.

Optimal Strategy

- Firstly, one can observe from Figure 4.23 that the optimal strategy is highly insensitive to the stage in the season.
- If the *advantage* at the start of $Q4$ is well in favour of the team, but the team starts with a very low initial energy reserve, it optimal to use up the remaining energy against a relatively stronger opponent. However, starting the game with moderate energy reserves, good advantage always calls for saving energy in $Q4$ (Figure 4.23a).
- When the relative strength is strongly in favour of the *team*, and the scores are level at the start of $Q4$, we can see from Figure 4.23b that it is indeed optimal to

save energy for future. If the *team* starts the game very low on energy, it is again optimal to save when scores are level after $Q3$.

- If the score-difference is largely in favour of the *field* after $Q3$, unless it is the case that the opponent is very weak compared to the *team*, it is optimal to save energy in $Q4$ (there is a sharp drop to zero, see Figure 4.23c).



(a) Ahead (b) Even (c) Behind

Figure 4.23: Optimal strategy at different stages of the season. Colour codes: $t =$ 5 , 15 , 40 , 60 , 75.

End Season

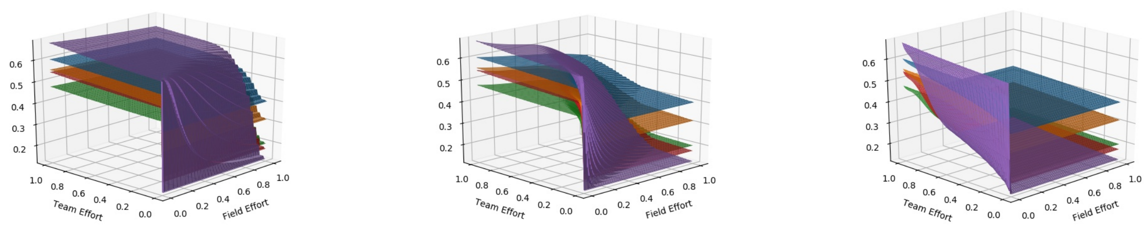
Similar to the base model, we once again focus attention to the last five games in the season in which, future games must be taken into consideration. Table 4.3 presents the number of strong opponents the *team* has yet to face, at the start of games in the last stages of the season.

Current Game	# Strong Opponents Remaining
75	3
76	3
77	2
78	1
79	0

Table 4.3: Number of Strong Opponents to face in remaining games

Expected Win Rate

- Figure 4.24 shows EWR at the start of the game, for games towards the end of the season. The variation of EWR across states is much larger compared to games earlier in the season.
- The combined effect of: (1) time in the season and (2) the number of strong opponents remaining is less obvious.



(a) Ahead (b) Even (c) Behind

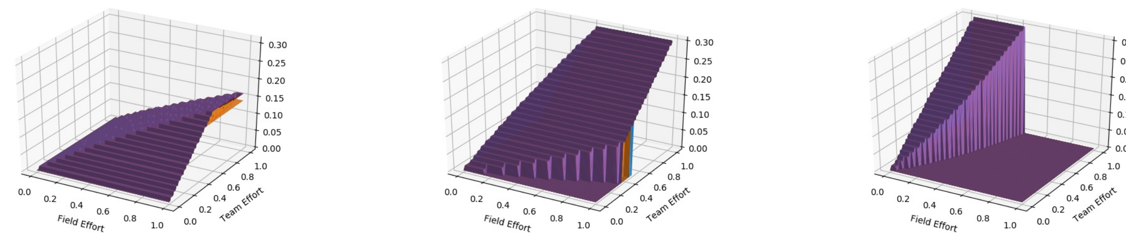
Figure 4.24: Expected Win Rate. Colour codes: $t = 75$, 76 , 77 , 78 , 79

Optimal Action in Q4

- We can observe from Figure 4.25 that the optimal action is highly insensitive to which game we consider even towards the end of the season.
- Moreover, the optimal strategy resembles closely with the optimal action at $Q4$ for earlier games in the season.

4.4.4 Comparison with Base Model

In this section, we discuss the similarities and differences in EWR and optimal strategies in each extension with respect to the base model. We adopt the following shorthand to refer to the different models: BM for "Base Model", FS for "Field Plays Optimal Strategy", WS for "Weak and Strong types of opponent" and finally AQ for "Team plays strategically in All Quarters".



(a) Ahead (b) Even (c) Behind

Figure 4.25: Optimal Strategy at the end of the season. Different shades denote different game (t), with the following colour codes: $t =$ 75 , 76 , 77 , 78 , 79

Expected Win Rate (EWR)

- In all cases, variability in *EWR* across states shrinks for games at the beginning of the season: initial conditions are inconsequential in a long season.

- Table 4.4 presents a detailed comparison of descriptive statistics for *EWR* at different stages of the season, across the four different models.

Model	min	max	mean	variance	min	max	mean	variance
BM	0.4950	0.6007	0.5494	0.0003	0.2486	0.8224	0.5408	0.0229
FS	0.5086	0.6147	0.5649	0.0003	0.2500	0.8242	0.5470	0.0235
AQ	0.5558	0.6910	0.6215	0.0003	0.2499	0.9588	0.5836	0.0218
WS	0.4473	0.6128	0.5559	0.0006	0.1263	0.6804	0.4570	0.0275

(a) **Games 10 to 70** (b) **Games 75 to 79**

Table 4.4: Comparison of *EWR* across the different models.

- The additional flexibility that the coach enjoys under AQ in terms of being able to act strategically during the first three quarters, allows the coach, when facing a team that has high energy, to decide to simply save all the energy. This flexibility translates into better EWR, as can be seen from the above table.

- The interplay between *advantage* and relative strength plays a key role in determining EWR in a similar fashion across the models. The effect (degree of rise or fall in EWR) is however different in each case.

Optimal Strategies

- The optimal strategies in $Q4$ remain highly insensitive to the time in the season, for all the different models.

- Table 4.5 compares the optimal strategies in different extensions with respect to the base model (BM), for the three advantage scenarios that the team might face at the start of Q4. Note that, the state space when the team plays strategically in all quarters (AQ) is higher-dimensional when compared to the other models.

Model	Advantage: Ahead	Advantage: Even	Advantage: Behind
BM	If the team starts with a very low initial energy reserve, it is optimal to use up the remaining energy against a relatively stronger opponent. However, if starting with moderate reserves, it is always always optimal save energy in Q4.	If the relative strength is strongly in favour of the team, it is optimal to save energy at Q4. However, if the team starts very low on energy, it is optimal to save in Q4.	Unless it is the case that the opponent starts with very low energy levels compared to the team, it is optimal to save energy in Q4 (there is a sharp drop to zero).
FS	No noticeable difference compared to BM.	If the relative strength is highly in favour of the field, the optimal action in BM is to just save energy, while in FS there are occasional spikes of energy usage.	No noticeable difference compared to BM.
AQ	Restriction to states representing final quarter decisions display no noticeable difference compared to BM.	Restriction to states representing final quarter decisions display no noticeable difference compared to BM.	Restriction to states representing final quarter decisions display no noticeable difference compared to BM.
WS	No noticeable difference compared to BM.	No noticeable difference compared to BM.	No noticeable difference compared to BM.

Table 4.5: Comparison of optimal strategies across the different models.

4.5 Concluding remarks

In this chapter, we have examined an interesting decision problem of a coach, who wishes to optimally manage the energy level of a pool of players in a competitive league. Given how players are exposed to frequent high-intensity movements during a game which can lead to acute and accumulated fatigue, coaches often consider holding back the intensity of play at the beginning of games and make an informed decision at a later stage based on the score-difference. In particular, something that coaches are known to frequently consider is how much to rest or limit minutes of star players in the team. However, not having the players put sufficient effort in a game could also prove to be costly. We have explored this trade-off between "saving energy for future games" and "winning current game" and the optimal energy dynamics late in the game, in a long season. We have shown that, indeed, saving energy is optimal throughout most of the season. The conclusion is checked for robustness under different extensions of the base model. Indeed, when a field of teams whose coaches employ similar strategies, it remains optimal to save energy in games throughout most of the season.

Since our focus was on effort dynamics late in the game, we resorted to allowing the coach to take a strategic decision once during every game. In future work, it would be interesting to study the problem where there are multiple decision making points in every game; and the coach needs to make a decision at every such point as a function of the score-difference at that point, the likelihood of winning the current game, and consideration for future games. Another interesting direction to explore is when every round of play of the game is a battle, making every individual game a multi-battle contest; and the winner of any given game is the one who wins (lets say) the majority of the battles. Since all the battles need not take place in every round (for example, if we consider a best of three battles contest, if a team wins the first two battles, it is declared the winner and the third battle is not fought at all), it might be interesting to study the energy dynamics in competitions with a series of such multi-battle contests.

4.6 Appendix C

4.6.1 Proofs

Proof of Proposition 4.2.1

Since, $V_{T+1}(s) = 0 \ \forall s \in S$, we have for all e^F, e^P and a:

$$V_T(e^P, e^F, a) = \max_x R(e^P, e^F, a, x)$$

Since, there is no value of saving in game T:

$$x_T = \operatorname*{argmax}_x R(e^P, e^F, a, x) = (1-\alpha)e^P$$

This implies, for all e^F, a and for all $e^P > \hat{e}^P$:

$$V_T(e^P, e^F, a) \quad \geq \quad V_T(\hat{e}^P, e^F, a) \qquad \text{(By Assumption 4.2.2)} \tag{4.6}$$

Now, applying backward induction, suppose it is true that for some $t \leq T$; for all e^F, a and for all $e^P > \hat{e}^P$,

$$V_t(e^P, e^F, a) \quad \geq \quad V_t(\hat{e}^P, e^F, a) \tag{4.7}$$

Next note that, $R(e^P, e^F, a, x)$ is increasing in a by definition when all other arguments are held fixed. This implies $V_t(e^P, e^F, a)$ is weakly increasing in a for fixed e^F and e^P. Then, by Assumption 4.2.3 we get,

$$\sum_a V_t(e^P, e^F, a)\eta(a|\ \alpha e^P, \beta e^F) \quad \geq \quad \sum_a V_t(\hat{e}^P, e^F, a)\eta(a|\ \alpha \hat{e}^P, \beta e^F) \tag{4.8}$$

Integrating out e^F yields,

$$\int \{\sum_a V_t(e^P, e^F, a)\eta(a|\ \alpha e^P, \beta e^F)\} g(de^F) \quad \geq \quad \int \{\sum_a V_t(\hat{e}^P, e^F, a)\eta(a|\ \alpha \hat{e}^P, \beta e^F)\} g(de^F) \tag{4.9}$$

Inequality (4.9) together with Assumption 4.2.1 then implies that for $r > r'$,

$$\int\int \{\sum_a V_t(e^P, e^F, a)\eta(a|\ \alpha e^P, \beta e^F)\} g(de^F)\mu(de^P|r)$$
$$\geq \tag{4.10}$$
$$\int\int \{\sum_a V_t(e^P, e^F, a)\eta(a|\ \alpha e^P, \beta e^F)\} g(de^F)\mu(de^P|r')$$

Recall that $s = (e^p, e^F, a)$. The value function at period $t-1$ is:

$$V_{t-1}(s) = \max_{x \leq (1-\alpha)e^P} \left[R(s,x) + \int\int \{\sum_a V_t(e^P, e^F, a)\eta(\,a \mid \alpha e^P, \beta e^F)\}\ g(de^F)\mu(de^P|r(x)) \right] \tag{4.11}$$

Recall that for any given e^P, any maximiser (x^*) of the expression inside the max operator in (4.11) must satisfy: $x^* + r(x^*) = (1-\alpha)e^P$. For any fixed s, we know that $R(s,x)$ is (weakly) increasing in x. Since for fixed e^F, a and $e^P > \hat{e}^P$, the argument that maximises the expression in (4.11) when the state is $\hat{e}^P$, is also available at state e^P, we have for fixed e^F, a:

$$V_{t-1}(e^P, e^F, a) \quad \geq \quad V_{t-1}(\hat{e}^P, e^F, a) \qquad \text{if,}\ \ e^P > \hat{e}^P \tag{4.12}$$

Thus by the Principle of Induction the proof follows. ■

Proof of Proposition 4.2.2

For initial states $s', s'' \in \bar{S}$ and any strategy σ_1, consider a strategy σ_2 to be as guaranteed by the statement in the proposition. Let us now define $X_k := V(s''_k) - V(s'_k) - 4k$. Since the reward accumulated at each period is at most 1, the following inequalities are satisfied for strategies σ_1 and σ_2 :

$$\left|E_{\sigma_1}\left[V(s'_{k+1} \mid s'_k)\right] - V(s'_k)\right| \leq 2$$
$$\left|E_{\sigma_2}\left[V(s''_{k+1} \mid s''_k)\right] - V(s''_k)\right| \leq 2$$

which implies $E_{\sigma_1,\sigma_2}\left[X_{k+1} \mid s'_k, s''_k\right] \leq X_k$.

Next, let τ be a random variable denoting the earliest $k \leq N$ such that $V(s''_k) \geq V(s'_k)$, if there is no such k, then let $\tau = N+1$. Invoking the optimal stopping theorem, we have:

$$E_{\sigma_1,\sigma_2}[X_\tau] \;\leq\; E_{\sigma_1,\sigma_2}[X_o] = V(s_0'') - V(s_0') \tag{4.13}$$

Now,

$$\begin{aligned} E_{\sigma_1,\sigma_2}\big[X_\tau\big] = \sum_{k=1}^{N} E_{\sigma_1,\sigma_2}\big[V(s_k'') - V(s_k') \;-\; 4k \;\mid\; \tau = k\big] \cdot P\big[\tau = k\big] + \\ E_{\sigma_1,\sigma_2}\big[V(s_{N+1}'') - V(s_{N+1}') - 4(N+1) \;\mid\; \tau = N+1\big] \cdot P\big[\tau = N+1\big] \end{aligned}$$

or,

$$\begin{aligned} E_{\sigma_1,\sigma_2}\big[X_\tau\big] \;&\geq\; \sum_{k=1}^{N}(0-4N)\cdot P\big[\tau = k\big] \;\;+\;\; \big[-\big(T-(N+1-1)\big)-4(N+1)\big]\cdot\varepsilon \\ &= -4N \;+\; \big[-(T-N)-4N-4\big]\cdot\varepsilon \\ &= -[T \;+\; 3N \;+\; 4]\cdot\varepsilon \;-\; 4N \end{aligned} \tag{4.14}$$

Using (4.14) in (4.13) and dividing both sides of the inequality by T the result follows. ■

4.6.2 Code used for Python simulations

```
#======= Python Packages used ========#
import numpy as np
from scipy.stats import binom
import math
import matplotlib.pyplot as plt
import time
#====================================#


#======== Preliminary HELPER FUNCTIONS that will be used in the Bellman optimization ==========#


## Define G = A + 30 where 'A' is advantage at the start of Q4.
## A ranges from -30 to 30 => G ranges from 0 to 60.
## Since I am going to use 'binom' from scipy.stats, this
## transformation will be useful. One could alternatively make
## the necessary transformation directly at appropriate places
## without defing this new variable. I had begun coding it
## this way and therefore stuck to it.




#===========================#
#      STAGE REWARD         #
#===========================#

def reward(G, y, x):
    N = 20
    if x == 0.0 and y == 0.0:
        x = 1.0
        y = 1.0
    if G == -10:
        return binom.pmf(20, N, x/(x+y))/2.0

    elif G == 10:
        ### P(X_4 > 0) + 1/2*P[X_4 = 0]
        return (1.0 - binom.pmf(0, N, x/(x+y))/2.0 )

    elif np.isin(G, np.linspace(-9, 9,19)):
        return (1.0 -  binom.cdf(10 - G, N, x/(x+y)) + binom.pmf(10 - G, N, x/(x+y))/2.0)

    elif np.isin(G, np.linspace(-30, -11,20)):
        return 0.0
    else:
        return 1.0



#====================================================================================================#
#   The following function makes Evaluation of Expected Value at t+1 cleaner and faster.            #
#   Takes as input P[:,:,:] and V[t+1, :,:,:] : the value array for period t+1.(used in the Bellman)#
#   OUTPUTS a 2D array.  Dimensions corresponding to E_f and E_p.                                    #
#   The (i, j)th cell value is the expected value where the expectation is taken over G.             #
#====================================================================================================#

def expected_value_given_Ef_and_Ep(x,p):
    (g,e_f, e_p) = x.shape
    output = np.zeros((e_f,e_p))
    for i in range(0,e_p):
        for j in range(0,e_f):
            output[j,i] = sum(x[:,j,i]*p[:,j,i])
    return(output)




### Effort-usage funtions

def team(x,alpha):
    return(alpha*x)
```

```
def field(x,beta):
    return(beta*x)


def effort(x,y):
    if y == 0:
        return(0)
    else:
        return(x/y)



#=====================================================================================================#
#                              Value Matrix and other array Initialization                            #
#=====================================================================================================#

T = 80
G = 61
num_of_states_E_f = 100    ##  number of effort_grid_points for the field
num_of_states_E_p = 100    ##  number of effort_grid_points for the team

## V = np.zeros((T, G, E_f, E_p))
V = np.zeros((T+2,G,num_of_states_E_f,num_of_states_E_p))
V_new = np.zeros((T+2,G,num_of_states_E_f,num_of_states_E_p))    ## will be used in Extension-I
strategy = np.zeros((T+2,G,num_of_states_E_f,num_of_states_E_p))
strategy_prop = np.zeros((T+2,G,num_of_states_E_f,num_of_states_E_p))
strategy_new = np.zeros((T+2,G,num_of_states_E_f,num_of_states_E_p))
strategy_new_prop = np.zeros((T+2,G,num_of_states_E_f,num_of_states_E_p))
opt_strategy_effort_index = np.zeros((T+2,G,num_of_states_E_f,num_of_states_E_p))
opt_strategy_effort_index_new = np.zeros((T+2,G,num_of_states_E_f,num_of_states_E_p))




#======================================================================================================#
#   Distribution of G as a function of (E_f, E_p)   : G follows Bin(60, f(E_p)/f(E_p)+g(E_f)) - 30        #
#======================================================================================================#

# effort lies between [0,1]
s_0 = 0.0
s_n = 1.0

alpha = 0.7/3.0    ## team spends 0.7*E_p in the first 3quarters
beta = 1/4.0       ## field spends 0.25*E_f every quarter
beta_new = 0.7/3.0    ## will be used when field plays strategically


### Effort level initialization
E_f = np.linspace(s_0, s_n, num = num_of_states_E_f)
E_p = np.linspace(s_0, s_n, num = num_of_states_E_p)



#======================================================================================================#
#                                       =: Probability cube :=                                          #
#                  given e_f and e_p, P[:,e_f, e_p ] gives the mass on each value of G.                #
#     This is going to be heavily used everytime while computing the expectation term in the Bellman   #
#======================================================================================================#


P = np.zeros((G, num_of_states_E_f, num_of_states_E_p))

for i in range(len(E_p)):
    i_used = int(np.around(3.0*alpha*i))
    for j in range(len(E_f)):
        if i == 0 :
            if j == 0:
                p = 0.5
            else :
                p = 0.0
        else:
            p = E_p[i_used]/(E_p[i_used]+field(E_f[j],3.0*beta))

        for g in range(0, 61):
            P[g,j, i] = binom.pmf(g, 60, p)
```

```
#====================================================================#
#                Evaluation of Value at time T                       #
#====================================================================#
## field spends 0.25*E_f every quarter.
## Team spends everything in the last quarter since V_T+1 = 0.


for i in range(len(E_p)):
    for j in range(len(E_f)):
        for g in range(0,61):
            V[T,g,j,i] = reward((g-30),(beta)*E_f[j], (1.0 - 3.0*alpha)*E_p[i])



####################################################################################################
#================================            BASE MODEL         =======================================#
####################################################################################################

#=====================================================================#
#                Backward Induction  (Bellman)                        #
#=====================================================================#

for t in range(T-1,0,-1):
    print (time.ctime(), " t = ",t)
    ## Expectation over G : the function returns a 2D array. cell (j,i) denotes expected value given E_f[j], E_p[i]
    expected_value_over_G = expected_value_given_Ef_and_Ep( V[t+1,:,:,:], P )
    for i in range(0,len(E_p)):
    #### E_p[loc_Q4] = effort available in Q4 (to use now VS save for later periods)
        loc_available_Q4 = int(i - np.around(3.0*alpha*i))
        for j in range(0, len(E_f)):
            for g in range(0, 61):
                Y = np.zeros(loc_available_Q4 + 1)
                for i_hat in range(0,loc_available_Q4 +1):       # i_hat loops over available effort grid
                    Y[i_hat] = reward(g-30,field(E_f[j],beta),E_p[i_hat]) +
                               [~ code contd.] (expected_value_over_G[:, (loc_available_Q4 - i_hat):].sum())*
                               [~ code contd.] (1/len(E_f))*(1/len(E_p[(loc_available_Q4 - i_hat):]))
                V[t,g,j,i] = np.max(Y)
                opt_strategy_effort_index[t,g,j,i] = np.argmax(Y)
                strategy[t,g,j,i] = E_p[np.argmax(Y)]
                strategy_prop[t,g,j,i] = effort(E_p[np.argmax(Y)], E_p[loc_available_Q4])

print('\n Voila!')



####################################################################################################
#========================== Extension I: Field Plays Optimal Strategy ===============================#
####################################################################################################

### Base Model Helper Functions Required.  The Base Model must be solved before running this codes for
### this extension, as the optimal strategies evaluated there will be used here.


#============================================================================================#
#                          =: New Probability cube :=                                        #
#============================================================================================#

P_new = np.zeros((G, num_of_states_E_f, num_of_states_E_p)) ## P = np.zeros((G, E_f, E_p))

for i in range(0,len(E_p)):
    i_used = int(np.around(3.0*alpha*i))
    for j in range(0, len(E_f)):
        j_used = int(np.around(3.0*beta_new*j))
        if i == 0 :
            if j == 0:
                p = 0.5
            else :
                p = 0.0
        else:
            p = float(E_p[i_used]/(E_p[i_used]+E_f[j_used]))

        for g in range(0, 61):
```

```
            P_new[g,j, i] = binom.pmf(g, 60, p)




#================================================================================================#
#                              Evaluation of New Value at time T                                  #
#================================================================================================#

for i in range(0,len(E_p)):
    for j in range(0,len(E_f)):
        for g in range(0, 61):
            V_new[T,g,j,i] = reward((g-30),(1.0 - 3.0*beta_new)*E_f[j], (1.0 - 3.0*alpha)*E_p[i])


##### Field spends 0.7*E_f in first three quarters and 0.3*E_f in the Q4. (since field plays the optimal strategy)
##### Team spends everything in the last quarter since V_T+1 = 0.


#=====================================================================#
#                        Backward Induction  (Bellman)                #
#=====================================================================#

for t in range(T-1,0,-1):
    print (time.ctime(), " t = ",t)
    ## Expectation over G : the function returns a 2D array. cell (j,i) denotes expected value given E_f[j], E_p[i]
    expected_value_over_G = expected_value_given_Ef_and_Ep( V_new[t+1,:,:,:], P_new )
    for i in range(0,len(E_p)):
    #### E_p[loc_Q4] = effort available in Q4 (to use now VS save for later periods)
        loc_available_Q4 = int(i - np.around(3.0*alpha*i))
        for j in range(0, len(E_f)):
            loc_available_Q4_field = int(j - np.around(3.0*beta_new*j))
            for g in range(0, 61):
                #### opt_strategy_effort_index comes from earlier optimization problem.
                field_strategy_effort_index = int(opt_strategy_effort_index[t,60-g,i,j])
                Y = np.zeros(loc_available_Q4 + 1)
                for i_hat in range(0,loc_available_Q4 +1):       # i_hat loops over available effort grid
                    Y[i_hat] = reward(g-30, E_f[field_strategy_effort_index], E_p[i_hat]) +
                               [~ code contd.] (expected_value_over_G[:, (loc_available_Q4 - i_hat):].sum())*
                               [~ code contd.] (1/len(E_f))*(1/len(E_p[(loc_available_Q4 - i_hat):]))
                V_new[t,g,j,i] = np.max(Y)
                opt_strategy_effort_index_new[t,g,j,i] = np.argmax(Y)
                strategy_new[t,g,j,i] = E_p[np.argmax(Y)]
                strategy_new_prop[t,g,j,i] = effort(E_p[np.argmax(Y)], E_p[loc_available_Q4])

print('\n Voila!')







##################################################################################################
#========================= Extension II: Strategic All Quarters      ==============================#
##################################################################################################




#========    Additional  HELPER FUNCTIONS that will be used in the Bellman optimization    ==========#

# j : field effort index (1/4th of it used in Q4, embedded in the reward function)
# i : team effort index (effort available in Q4 (state variable))
# a : advantage (state variable)
# energy : E_f and E_p already defined earlier in the code which is being used here.
# value_Q3 : input giving value at Q3 (numbers come from calculations done in t+1 )


def max_bellman_Q4(j, i, adv, value_Q3):
    val = np.zeros(i + 1)
    for r in range(0, i+1):
        val[r] = reward(adv , (1/4.0)*E_f[j], E_p[i - r]) + (value_Q3[:,r:].sum())*(1/len(E_f)*(1/len(E_p[r:])))
    (value, opt_saving_index) = (np.max(val),np.argmax(val))
    return((value,opt_saving_index))
```

```
def max_bellman_Q3(j, i, value_Q4):
    val = np.zeros(i + 1)
    for x in range(0, i+1):
        val[x] = sum(value_Q4[i-x,:]*P[:,j,x])
    (value_Q3, opt_effort_index) = (np.max(val),np.argmax(val))
    return((value_Q3,opt_effort_index))



#==========================================================================================================#
#                            Value Matrix and other array Initialization                                   #
#==========================================================================================================#

T = 80
G = 61
num_of_states_E_f = 100    ### number of effort_grid_points
num_of_states_E_p = 100    ### number of effort_grid_points


V_Q3   = np.zeros((T+2, num_of_states_E_f, num_of_states_E_p))
Opt_Q3 = np.zeros((T+2, num_of_states_E_f, num_of_states_E_p))

V_Q4_interim   = np.zeros((T+2, num_of_states_E_f, num_of_states_E_p, G))
Opt_Q4_interim = np.zeros((T+2, num_of_states_E_f, num_of_states_E_p, G))



## Initializing Effort variables.
s_0 = 0.0
s_n = 1.0

beta = 3/4.0  #### field spends beta*E_f in first three quarters

E_f = np.linspace(s_0, s_n, num = num_of_states_E_f)
E_p = np.linspace(s_0, s_n, num = num_of_states_E_p)




#=========================================================================================================#
#                      State (ADVANTAGE) transition Probabilities at the end of Q3                        #
#                P[:, j, i ] gives conditional distribution of ADV given effort levels.                   #
#                           where,    P[a, j, i ]  =  P[a | 3/4*E_f[j], E_p[i]]                           #
#=========================================================================================================#

P = np.zeros((G, num_of_states_E_f, num_of_states_E_p)) ## P = np.zeros((G, E_f, E_p))

for i in range(0,len(E_p)):
    for j in range(0, len(E_f)):
        if i == 0 :
            if j == 0:
                p = 0.5
            else :
                p = 0.0
        else:
            p = E_p[i]/(E_p[i] + (beta*E_f[j]))   ######## FIELD uses 3/4th of it's effort in first three quarters  ####

        for g in range(0, 61):
            P[g,j, i] = binom.pmf(g, 60, p)
```

```
#==========================================================================================#
#                                  Backward Induction  (Bellman)                           #
#==========================================================================================#

for t in range(T,0,-1):
    print (time.ctime(), " t = ",t)
    ### Loop over FIELD effort levels
    for j in range(0, len(E_f)):
        ### next two loops - visits to ALL possible Q4 states
        for k in range(0, len(E_p)):
            for a in range(0,61):
                ### Maximisation at Q4 for each possible state (e_P,a) for fixed e_F (j)
                (val_Q4_interim , opt_saving_Q4_index) = max_bellman_Q4(j, k, a-30, V_Q3[t+1,:,:])
                V_Q4_interim[t,j,k,a]    =  val_Q4_interim
                Opt_Q4_interim[t,j,k,a]  =  E_p[opt_saving_Q4_index]  ### remember it is optimal SAVING (not energy used in Q4)
        ### Loop over TEAM effort levels
        for i in range(0,len(E_p)):
            (val_Q3 , opt_used_Q3_index) = max_bellman_Q3(j, i, V_Q4_interim[t,j,:(i+1),:])
            V_Q3[t,j,i]     =  val_Q3
            Opt_Q3[t,j,i]   =  E_p[opt_used_Q3_index]

print('\n Voila!')
```

```
####################################################################################################
#========================= Extension III: Strong and Weak opponents     ===========================#
####################################################################################################


### Base Model Helper Functions Required.

T = 80
G = 61
num_of_states_E_f = 100    ###### number of effort_grid_points
num_of_states_E_p = 100    ###### number of effort_grid_points
random.seed(1234)
opponent_strength_in_game = np.random.randint(2, size=T+2)   ### 0: LOW and 1: HIGH


V = np.zeros((T+2,G, num_of_states_E_f, num_of_states_E_p))
V_new = np.zeros((T+2,G, num_of_states_E_f, num_of_states_E_p))
strategy = np.zeros((T+2,G, num_of_states_E_f, num_of_states_E_p))
strategy_prop = np.zeros((T+2,G, num_of_states_E_f, num_of_states_E_p))
strategy_new = np.zeros((T+2,G, num_of_states_E_f, num_of_states_E_p))
strategy_new_prop = np.zeros((T+2,G, num_of_states_E_f, num_of_states_E_p))
opt_strategy_effort_index = np.zeros((T+2,G, num_of_states_E_f, num_of_states_E_p))
opt_strategy_effort_index_new = np.zeros((T+2,G, num_of_states_E_f, num_of_states_E_p))



P = np.zeros((G, num_of_states_E_f, num_of_states_E_p)) ## P = np.zeros((G, E_f, E_p))

for i in range(0,len(E_p)):
    for j in range(0, len(E_f)):
        if i == 0 :
            if j == 0:
                p = 0.5
            else :
                p = 0.0
        else:
            p = float(team(E_p[i],alpha)/(team(E_p[i],alpha)+field(E_f[j],beta)))

        for g in range(0, 61):
            P[g,j, i] = binom.pmf(g, 60, p)


for i in range(0,len(E_p)):
    for j in range(0,len(E_f)):
        for g in range(0, 61):
            V[T,g,j,i] = reward((g-30),(beta)*E_f[j], (1.0 - 3.0*alpha)*E_p[i])
```

```
#==========================================================================================#
#                                   Backward Induction  (Bellman)                          #
#==========================================================================================#

for t in range(T-1,0,-1):
    print (time.ctime(), " t = ",t)
    ## Expectation over G : the function returns a 2D array. cell (j,i) denotes expected value given E_f[j], E_p[i]
    expected_value_over_G = expected_value_given_Ef_and_Ep( V[t+1,:,:,:], P )
    for i in range(0,len(E_p)):
        #### E_p[loc_Q4] = effort available in Q4 (to use now VS save for later periods)
        loc_available_Q4 = int(i - np.around(3.0*alpha*i))
        for j in range(0, len(E_f)):
            for g in range(0, 61):
                Y = np.zeros(loc_available_Q4 + 1)
                if opponent_strength_in_game[t+1] == 0: ### WEAK Type TOMORROW ###
                    for i_hat in range(0,loc_available_Q4 +1):      # i_hat loops over available effort grid
                        Y[i_hat] = reward(g-30,field(E_f[j],beta),E_p[i_hat]) +
                        [~ code contd.] (expected_value_over_G[:int(len(E_f)/2), (loc_available_Q4 - i_hat):].sum())*
                        [~ code contd.] (2.0/len(E_f))*(1/len(E_p[(loc_available_Q4 - i_hat):]))
                else:  ### STRONG Type TOMORROW ###
                    for i_hat in range(0,loc_available_Q4 +1):      # i_hat loops over available effort grid
                        Y[i_hat] = reward(g-30,field(E_f[j],beta),E_p[i_hat]) +
                        [~ code contd.] (expected_value_over_G[int(len(E_f)/2):, (loc_available_Q4 - i_hat):].sum())*
                        [~ code contd.] (2.0/len(E_f))*(1/len(E_p[(loc_available_Q4 - i_hat):]))
                V[t,g,j,i] = np.max(Y)
                opt_strategy_effort_index[t,g,j,i] = np.argmax(Y)
                strategy[t,g,j,i] = E_p[np.argmax(Y)]
                strategy_prop[t,g,j,i] = effort(E_p[np.argmax(Y)], E_p[loc_available_Q4])

print('\n Voila!')




##================================================##
##          Example Code For The Graphs           ##
##================================================##


## following imports necessary
from numpy import *
%matplotlib notebook
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D



## The same code structure works for generating all graphs
## simply by replacing the Value array as well as other parameters appropriately.
## I therefore do not include it here.

t1 = 10
t2 = 40
t3 = 50
t4 = 60
t5 = 70

g_high = 37
g_mid = 30
g_low = 23

x,y = mgrid[0:len(E_p), 0:len(E_p)]

f_mid_t1 = V[t1,g_mid,x,y]/(T - t1 +1)
f_mid_t2 = V[t2,g_mid,x,y]/(T - t2 +1)
f_mid_t3 = V[t3,g_mid,x,y]/(T - t3 +1)
f_mid_t4 = V[t4,g_mid,x,y]/(T - t4 +1)
f_mid_t5 = V[t5,g_mid,x,y]/(T - t5 +1)


fig = plt.figure()
ax_mid = fig.add_subplot(111,projection = '3d')
```

```
ax_mid.plot_surface(E_f[x],E_p[y],f_mid_t1,rstride=1, cstride=1)
ax_mid.plot_surface(E_f[x],E_p[y],f_mid_t2,rstride=1, cstride=1)
ax_mid.plot_surface(E_f[x],E_p[y],f_mid_t3,rstride=1, cstride=1)
ax_mid.plot_surface(E_f[x],E_p[y],f_mid_t4,rstride=1, cstride=1)
ax_mid.plot_surface(E_f[x],E_p[y],f_mid_t5,rstride=1, cstride=1)

ax_mid.set_xlabel('\nField Effort')
ax_mid.set_ylabel('\nTeam Effort')


##########################################################################################################################
##########################################################################################################################
##########################################################################################################################
```

Bibliography

Abdulkadiroğlu A, Sönmez T (1999) House allocation with existing tenants. Journal of Economic Theory 88(2):233–260

Andersson T, Ehlers L (2019) Assigning refugees to landlords in sweden: Efficient stable maximum matchings. The Scandinavian Journal of Economics

Andersson T, Kratz J (2020) Pairwise kidney exchange over the blood group barrier. The Review of Economic Studies 87(3):1091–1133

Arkes J, Martinez J (2011) Finally, evidence for a momentum effect in the nba. Journal of Quantitative Analysis in Sports 7(3)

Åslund O, Rooth DO (2007) Do when and where matter? initial labour market conditions and immigrant earnings. The Economic Journal 117(518):422–448

Åslund O, Östh J, Zenou Y (2010) How important is access to jobs? old question—improved answer. Journal of Economic Geography 10(3):389–422

Aziz H, De Keijzer B (2012) Housing markets with indifferences: A tale of two mechanisms. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 26

Bade S (2019) Matching with single-peaked preferences. Journal of Economic Theory 180:81–99

Barbera S (2001) An introduction to strategy-proof social choice functions. Social Choice and Welfare 18(4):619–653

Baye MR, Kovenock D, De Vries CG (1996) The all-pay auction with complete information. Economic Theory 8(2):291–305

Berger J, Pope D (2011) Can losing lead to winning? Management Science 57(5):817–827

Berkes F (1992) Success and failure in marine coastal fisheries of turkey. Making the commons work: Theory, practice, and policy pp 161–182

Bogomolnaia A, Moulin H (2002) A simple random assignment problem with a unique solution. Economic Theory 19(3):623–636

Bogomolnaia A, Deb R, Ehlers L (2005) Strategy-proof assignment on the full preference domain. Journal of Economic Theory 123(2):161–186

Cattelan M, Varin C, Firth D (2013) Dynamic bradley–terry modelling of sports tournaments. Journal of the Royal Statistical Society: Series C (Applied Statistics) 62(1):135–150

Chun Y, Park B (2017) A graph theoretic approach to the slot allocation problem. Social Choice and Welfare 48(1):133–152

Clarke EH (1971) Multipart pricing of public goods. Public choice pp 17–33

Crès H, Moulin H (2001) Scheduling with opting out: Improving upon random priority. Operations Research 49(4):565–577

Crowder M, Dixon M, Ledford A, Robinson M (2002) Dynamic modelling and prediction of english football league matches for betting. Journal of the Royal Statistical Society: Series D (The Statistician) 51(2):157–168

Damm AP (2014) Neighborhood quality and labor market outcomes: Evidence from quasi-random neighborhood assignment of immigrants. Journal of Urban Economics 79:139–166

Dubins LE, Freedman DA (1981) Machiavelli and the gale-shapley algorithm. The American Mathematical Monthly 88(7):485–494

Ehlers L (2002) Coalitional strategy-proof house allocation. Journal of Economic Theory 105(2):298–317

Erdil A, Ergin H (2008) Whats the matter with tie-breaking. Improving effi ciency in school choice, forthcoming in American Economic Review

Erdil A, Ergin H (2017) Two-sided matching with indifferences. Journal of Economic Theory 171:268–292

Ergin HI (2002) Efficient resource allocation on the basis of priorities. Econometrica 70(6):2489–2497

Friedman L (1958) Game-theory models in the allocation of advertising expenditures. Operations research 6(5):699–709

Gale D, Shapley LS (1962) College admissions and the stability of marriage. The American Mathematical Monthly 69(1):9–15

Gale D, Sotomayor M (1985) Ms. machiavelli and the stable matching problem. The American Mathematical Monthly 92(4):261–268

Geyer A, Hanke M, Weissensteiner A (2009) Life-cycle asset allocation and consumption using stochastic linear programming. Journal of Computational Finance 12(4):29–50

Ghosh S, Long Y, Mitra M (2020) Prior-free online mechanisms for queueing with arrivals. Economic Theory pp 1–30

Gibbard A (1973) Manipulation of voting schemes: a general result. Econometrica: journal of the Econometric Society pp 587–601

Gómez MÁ, Lago-Peñas C, Pollard R (2013) Situational variables. In: Routledge handbook of sports performance analysis, Routledge, pp 277–287

Green J, Laffont JJ (1977) Characterization of satisfactory mechanisms for the revelation of preferences for public goods. Econometrica: Journal of the Econometric Society pp 427–438

Hatfield JW, Kojima F (2009) Group incentive compatibility for matching with contracts. Games and Economic Behavior 67(2):745–749

Hougaard JL, Moreno-Ternero JD, Østerdal LP (2014) Assigning agents to a line. Games and Economic Behavior 87:539–553

Hurwicz L (1973) The design of mechanisms for resource allocation. The American Economic Review 63(2):1–30, URL http://www.jstor.org/stable/1817047

Irving RW (1994) Stable marriage and indifference. Discrete Applied Mathematics 48(3):261–272

Irving RW, Manlove DF, Scott S (2008) The stable marriage problem with master preference lists. Discrete Applied Mathematics 156(15):2959–2977

Jackson MO, Nicolo A (2004) The strategy-proof provision of public goods under congestion and crowding preferences. Journal of Economic Theory 115(2):278–308

Jones W, Teytelboym A (2017) The international refugee match: A system that respects refugees' preferences and the priorities of states. Refugee Survey Quarterly 36(2):84–109

Jones W, Teytelboym A (2018) The local refugee match: Aligning refugees' preferences with the capacities and priorities of localities. Journal of Refugee Studies 31(2):152–178

Kennes J, Monte D, Tumennasan N (2014) The day care assignment: A dynamic matching problem. American Economic Journal: Microeconomics 6(4):362–406

Klaus B (2001) Coalitional strategy-proofness in economies with single-dipped preferences and the assignment of an indivisible object. Games and Economic Behavior 34(1):64–82

Klaus B, Peters H, Storcken T (1998) Strategy-proof division with single-peaked preferences and individual endowments. Social Choice and Welfare 15(2):297–311

Klumpp T, Konrad KA, Solomon A (2019) The dynamics of majoritarian blotto games. Games and Economic Behavior 117:402–419

Korpela V, Lombardi M, Saulle R (2021) An implementation approach to rotation programs. Available at SSRN 3833533

Kraft H, Munk C (2011) Optimal housing, consumption, and investment decisions over the life cycle. Management Science 57(6):1025–1041

Lago C (2009) The influence of match location, quality of opposition, and match status on possession strategies in professional association football. Journal of sports sciences 27(13):1463–1469

Lazear EP, Rosen S (1981) Rank-order tournaments as optimum labor contracts. Journal of political Economy 89(5):841–864

Li S (2017) Obviously strategy-proof mechanisms. American Economic Review 107(11):3257–87

Ma J (1994) Strategy-proofness and the strict core in a market with indivisibilities. International Journal of Game Theory 23(1):75–83

Mandal P, Roy S (2021) Strategy-proof allocation of indivisible goods when preferences are single-peaked. Tech. rep., University Library of Munich, Germany

Maniquet F, Sprumont Y (1999) Efficient strategy-proof allocation functions in linear production economies. Economic Theory 14(3):583–595

Manlove D (2013) Algorithmics of matching under preferences, vol 2. World Scientific

Manlove DF (2002) The structure of stable marriage with indifference. Discrete Applied Mathematics 122(1-3):167–181

Manlove DF, Irving RW, Iwama K, Miyazaki S, Morita Y (2002) Hard variants of stable marriage. Theoretical Computer Science 276(1-2):261–279

Martén L, Hainmueller J, Hangartner D (2019) Ethnic networks can foster the economic integration of refugees. Proceedings of the National Academy of Sciences 116(33):16280–16285

McVitie DG, Wilson LB (1970) Stable marriage assignment for unequal sets. BIT Numerical Mathematics 10(3):295–309

Merton R (1971) Optimum consumption and portfolio-rules in a continuous-time framework. Journal of Economic Theory (December 1971)

Moulin H (1980) On strategy-proofness and single peakedness. Public Choice 35(4):437–455

Moulin H (1994) Serial cost-sharing of excludable public goods. The Review of Economic Studies 61(2):305–325

Moulin H (2014) Cooperative microeconomics: a game-theoretic introduction, vol 313. Princeton University Press

Moulin H (2017) One-dimensional mechanism design. Theoretical Economics 12(2):587–619

Myerson RB (1981) Optimal auction design. Mathematics of operations research 6(1):58–73

Myerson RB, Satterthwaite MA (1983) Efficient mechanisms for bilateral trading. Journal of economic theory 29(2):265–281

Neiman T, Loewenstein Y (2011) Reinforcement learning in professional basketball players. Nature communications 2(1):1–8

Osterman P (1994) How common is workplace transformation and who adopts it? Ilr Review 47(2):173–188

Ostrom E (1990) Governing the commons: The evolution of institutions for collective action. Cambridge university press

O'Donoghue P (2009) Interacting performances theory. International Journal of Performance Analysis in Sport 9(1):26–46

Pápai S (2000) Strategyproof assignment by hierarchical exchange. Econometrica 68(6):1403–1433

Pereyra JS (2013) A dynamic school choice model. Games and economic behavior 80:100–114

Pycia M, Ünver MU (2017) Incentive compatible allocation and exchange of discrete resources. Theoretical Economics 12(1):287–329

Roth AE (1982) Incentive compatibility in a market with indivisible goods. Economics letters 9(2):127–132

Roth AE, Sotomayor MAO (1990) Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis. Econometric Society Monographs, Cambridge University Press, DOI 10.1017/CCOL052139015X

Roth AE, Sönmez T, Ünver MU (2004) Kidney exchange. The Quarterly journal of economics 119(2):457–488

Roth AE, Sönmez T, Ünver MU (2005) Pairwise kidney exchange. Journal of Economic theory 125(2):151–188

Rue H, Salvesen O (2000) Prediction and retrospective analysis of soccer matches in a league. Journal of the Royal Statistical Society: Series D (The Statistician) 49(3):399–418

Sampaio J, Lago C, Casais L, Leite N (2010) Effects of starting score-line, game location, and quality of opposition in basketball quarter score. European Journal of Sport Science 10(6):391–396

Sampaio J, Ibáñez S, Lorenzo A (2013) Applied sports performance analysis. basketball. T McGarry, P O'Donoghue J, and J Sampaio (Eds), Routledge Handbook of Sports Performance Analysis pp 357–358

Samuelson PA (1969) Lifetime portfolio selection by dynamic stochastic programming. The Review of Economics and Statistics 51(3):239–246, URL http://www.jstor.org/stable/1926559

Satterthwaite MA (1975) Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. Journal of economic theory 10(2):187–217

Satterthwaite MA, Sonnenschein H (1981) Strategy-proof allocation mechanisms at differentiable points. The Review of Economic Studies 48(4):587–597

Sela A, Erez E (2013) Dynamic contests with resource constraints. Social Choice and Welfare 41(4):863–882

Shapley L, Scarf H (1974) On cores and indivisibility. Journal of mathematical economics 1(1):23–37

Sönmez T (1999) Strategy-proofness and essentially single-valued cores. Econometrica 67(3):677–689

Stojanović E, Stojiljković N, Scanlan AT, Dalbo VJ, Berkelmans DM, Milanović Z (2018) The activity demands and physiological responses encountered during basketball match-play: a systematic review. Sports Medicine 48(1):111–135

Svensson LG (1994) Queue allocation of indivisible goods. Social Choice and Welfare 11(4):323–330

Svensson LG (1999) Strategy-proof allocation of indivisible goods. Social Choice and Welfare 16(4):557–567

Taylor JB, Mellalieu SD, James N, Shearer DA (2008) The influence of match location, quality of opposition, and match status on technical performance in professional association football. Journal of Sports Sciences 26(9):885–895

Thorpe RT, Atkinson G, Drust B, Gregson W (2017) Monitoring fatigue status in elite team-sport athletes: implications for practice. International journal of sports physiology and performance 12(s2):S2–27

Tullock G (1967) The welfare costs of tariffs, monopolies, and theft. Economic Inquiry 5(3):224–232

Vickrey W (1961) Counterspeculation, auctions, and competitive sealed tenders. The Journal of finance 16(1):8–37

Yu J, Zhang J (2020) A market design approach to job rotation. Games and Economic Behavior 120:180–192

Zariphopoulou T (1994) Consumption-investment models with constraints. SIAM Journal on Control and Optimization 32(1):59–85