

THREE FUNDAMENTAL PILLARS OF
DECISION-CENTERED TEAMWORK

by

Leandro Soriano Marcolino

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

August 2016

Copyright 2016

Leandro Soriano Marcolino

To my father

Acknowledgements

When a chess grandmaster performs a brilliant move, she may think of herself as a genius. But that move would never have been made without the lessons of her previous mentors and the pressure of her opponent. When an athlete wins a gold medal, she may think of herself as superior. But nothing would have been achieved without the training of her coaches and the support of family and friends. When a scientist has a breakthrough, she may think of it as her greatest achievement. However, that breakthrough is the result of her standing “in the shoulder of giants”, as Newton brilliantly realized.

No achievement in life is performed alone. We are a product of our interactions with our family, our friends, our masters, our colleagues and collaborators. We learn and grow with everyone around us, no matter if they are intentionally teaching us or not. We survive thanks to everyone around us, no matter if they are playing as friends or as antagonists. Life is a product of our struggles, and of the hands that pull us up when we fall, the hands that point the way forward when we are lost, the hands that push us ahead when we stop.

It is impossible to cite names without being unfair. It is also impossible to write an acknowledgment without citing names. The list of everyone that I own my gratitude would perhaps grow larger than this thesis itself. Hence, excuse my unfairness by first writing the names of my masters. I thank my advisor, Milind Tambe, for guiding me in those five years, not only in my research, but also in my formation as a professional researcher. I must also thank my previous advisors, Hitoshi Matsubara and Luiz Chaimowicz, who helped in shaping me as a researcher, and in preparing me to start my PhD. I also thank the members of the committee, Gaurav Sukhatme, William Swartout, Craig Knoblock and Nicholas Weller, for their time and guidance. In particular, Milind, Gaurav and William went beyond their usual duties to guide me in securing an academic position.

I also must thank my students, who believed in me to guide their work: Vaishnavh Nagarajan, Aravind Lakshminarayanan, Samori Price, Boian Kolev, Álvaro Souza, Douglas Chen. They taught me so much, I would be glad if they learned with me at least half of what I myself learned from them.

I thank my friends from the School of Architecture: Sreerag Palangat Veetil, Evangelos Pantazis, David Gerber. They gave me new and exciting ways to think about my own research, besides very interesting discussions and conversations.

I thank my collaborators, and the members of the AAMAS community. In particular, I thank Ariel Procaccia, Tuomas Sandholm, Maria Gini, Shimon Whiteson, Albert Xin Jiang, Chris Kiekintveld, Emma Bowring, William Haskell, Michael Wooldridge; who helped me in my research, and in finding a position to start my academic career.

Of course I must thank my wife for being with me throughout this journey, far from her family and friends of her own country, just to offer me her love and support. I also thank my family, who always support me, and care about me, no matter where I am.

For all the names that I failed to list, I offer you my sincere gratitude, in exchange of your forgiveness...

The end is nothing; the road is all. (Willa Cather)

Table of Contents

Acknowledgements	iii
List Of Figures	ix
List Of Tables	xiv
Abstract	xv
I Introduction and Background	1
Chapter 1 INTRODUCTION	2
1.1 Summary of Contributions	4
1.2 Guide to thesis	8
Chapter 2 BACKGROUND AND RELATED WORK	9
2.1 Social Choice	9
2.2 Teamwork	12
2.3 Social Science	13
2.4 Machine Learning	15
2.5 Team Assessment	15
2.6 Other Multiple Algorithms Techniques	17
2.7 Social Networks	17
II Agent Selection	20
Chapter 3 DIVERSITY BEATS STRENGTH?	21
3.1 Introduction	21
3.2 Methodology	23
3.3 Results	29
3.4 Detailed Study: Why a Team of Diverse Agents Perform Better?	35
3.5 Conclusion and Discussions	42

Chapter 4	GIVE A HARD PROBLEM TO A DIVERSE TEAM	44
4.1	Model for Analysis of Diversity in Teams	45
4.2	Experimental Analysis	53
4.3	Conclusion and Discussions	60
Chapter 5	SO MANY OPTIONS, BUT WE NEED THEM ALL	63
5.1	Introduction	63
5.2	Design Domains	65
5.3	Agent Teams for Design Problems	68
5.4	Synthetic Experiments	83
5.5	Experiments in Architectural Design	85
5.6	Discussion	90
5.7	Conclusion	96
III	Aggregation of Opinions	98
Chapter 6	RANKED VOTING	99
6.1	Introduction	99
6.2	Team Generation	101
6.3	Ranked Voting	101
6.4	PSINET	104
6.5	Results	106
6.6	Conclusion	112
Chapter 7	SIMULTANEOUS INFLUENCING AND MAPPING	115
7.1	Introduction	115
7.2	Influencing and Mapping	116
7.3	Results	122
7.4	Discussion	132
7.5	Conclusion	133
IV	Team Assessment	135
Chapter 8	EVERY TEAM DESERVES A SECOND CHANCE	136
8.1	Introduction	136
8.2	Prediction Method	139
8.3	Theory	142
8.4	Results	154
8.5	Analysis of Coefficients	168
8.6	Discussion	173
8.7	Conclusion	177

V	Discussions and Conclusions	179
Chapter 9	CONCLUSIONS	180
	References	183
	Appendices	199
Appendix A	SIMULTANEOUS INFLUENCING AND MAPPING: ADDITIONAL RESULTS	200
	A.1 Results for each network	200
	A.2 Additional results for power law distribution	200
Appendix B	EVERY TEAM DESERVES A SECOND CHANCE: ADDITIONAL RESULTS	219

List Of Figures

1.1	Domains where I explored decision-centered teamwork.	5
2.1	“Classical” voting model. Each agent has a noisy perception of the truth, or correct outcome. Hence, its vote is influenced by the correct outcome. .	10
3.1	1200 random teams of 4 agents.	30
3.2	Histogram of the agents, using real data.	33
3.3	Expected size of the set of agents that vote for the winning move, with 6 agents and no opening database.	34
3.4	Results in the Computer Go domain. The error bars show the confidence interval, with 99% of significance.	36
3.5	First example, the diverse team plays as white without the opening database against Fuego. White wins by resignation.	38
3.6	Second example, the diverse team plays as white without the opening database against Fuego. White wins by resignation.	40
3.7	Third example, the diverse team plays as white with the opening database against Fuego. White wins by resignation.	41
4.1	Comparing diverse and uniform when uniform also increases d_m	53
4.2	p_{best} of a diverse team as the number of agents increases.	54
4.3	Winning rate in the real Computer Go system.	57
4.4	Winning rates for 4 and 6 agents teams.	57
4.5	Histograms of agents for different board sizes.	59
4.6	Verifying the assumptions in the real system.	60
5.1	A parametric design of a building, showing two parameters: $X1$ and $Y1$. .	66
5.2	Illustrative example of the probability distribution functions of two diverse agents.	74
5.3	Illustrative example of Equation 5.1. Here I show 6 agents ($n := 6$) with 2 preferred actions each ($r := 2$). Each action is in the list of preferences of 4 agents ($k := 4$). As an example, I mark with a dashed circle one of the actions, a_2	75
5.4	Illustrative example of the probability distribution functions of two agents with multiple ${}_j\mathbf{Good}^i$ sets.	80

5.5	Percentage of optimal solutions found by uniform teams as $\max \lambda^+$ grows. Note how $\max \lambda^+ := 0$ gives the best result.	83
5.6	Percentage of optimal solutions as the number of agents grows. The uniform teams <i>decrease</i> in performance, while multiple variations of diverse teams improve, but with diminishing returns.	84
5.7	Percentage of optimal solutions found by large teams of diverse agents. . .	85
5.8	Parametric designs with increasing complexity used in our experiments. .	86
5.9	Percentage of optimal solutions of each system. The teams find a much larger percentage than the individual agents.	89
5.10	Some building designs generated by the teams.	91
5.11	Additional analysis. I show here that many falsely reported optimal solutions are eliminated by the team of agents, and also that the teams provide a large number of optimal solutions to the designer.	92
5.12	All the optimal solutions in the objectives space.	93
6.1	Winning rates for Diverse (continuous line) and Uniform (dashed line), for a variety of team sizes, using the plurality voting rule.	107
6.2	Winning rate of Fuego and of the parametrized agents.	107
6.3	Evaluation of the diversity of the parametrized agents, and the fraction of states in which all of them have a low probability of playing the optimal action. The error bars show 99% confidence intervals.	109
6.4	Winning rates for Diverse (continuous line) and Uniform (dashed line), for a variety of team sizes and voting rules.	110
6.5	All voting rules, for Diverse with 5 agents, using the new ranking methodology.	111
6.6	Comparison on BTER graphs	111
6.7	One of the friendship based social network of homeless people visiting My Friend's Place	113
6.8	Solution Quality for Real World Networks	113
7.1	A graph where the traditional greedy algorithm has arbitrarily low performance.	119
7.2	Results of 4 real world networks across many interventions, for $p = 0.5$ and $\varphi = 0.5$ (uniform distribution).	124
7.3	Results of Influence and Knowledge for different teaching and influence probabilities (uniform distribution).	126
7.4	Regret for different teaching and influence probabilities (uniform distribution). Lower results are better.	127
7.5	Results of 4 real world networks across many interventions, for $p = 0.5$ and $\varphi = 0.5$ (power law distribution).	129
7.6	Influence and Knowledge for different teaching and influence probabilities (power law distribution).	131

7.7	Regret for different teaching and influence probabilities (power law distribution). Lower results are better.	131
8.1	“Classical” voting model. Each agent has a noisy perception of the truth, or correct outcome. Hence, its vote is influenced by the correct outcome. .	144
8.2	My main model. I assume that the subset of agents that decided the action of the team at each world state \mathbf{H}_i determines whether the team will be successful or not (W).	147
8.3	Winning rates of the three teams, under four different board sizes.	155
8.4	Performance metrics over all turns of 9x9 Go games.	158
8.5	ROC curves, analyzing different thresholds in 9x9 Go.	160
8.6	AUC for <i>diverse</i> , <i>uniform</i> and <i>intermediate</i> teams, in 9x9 Go.	161
8.7	ROC curves for <i>diverse</i> and <i>uniform</i> , for different board sizes.	162
8.8	AUC for different teams and board sizes, organized by teams.	163
8.9	AUC for different teams and board sizes, organized by board sizes.	164
8.10	Differences in prediction quality for the <i>diverse</i> and <i>uniform</i> teams.	164
8.11	Performance metrics over all turns of 9x9 and 21x21 Go games.	166
8.12	Comparison of prediction quality with the full and reduced representation.	167
8.13	Performance metrics over a set of items in ensemble system classification (full feature vector).	169
8.14	Performance metrics over a set of items in ensemble system classification (reduced feature vector).	170
8.15	Normalized coefficients of all teams and board sizes, organized by board sizes.	172
8.16	Normalized coefficients of all teams and board sizes, organized by teams.	174
A.1	Results for network A across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming uniform distribution.	201
A.2	Results of Influence and Knowledge in network A for different teaching and influence probabilities, assuming uniform distribution.	202
A.3	Regret in network A for different teaching and influence probabilities, assuming uniform distribution.	202
A.4	Results for network B across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming uniform distribution.	203
A.5	Results of Influence and Knowledge in network B for different teaching and influence probabilities, assuming uniform distribution.	204
A.6	Regret in network B for different teaching and influence probabilities, assuming uniform distribution.	204
A.7	Results for Facebook network across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming uniform distribution.	205

A.8 Results of Influence and Knowledge in Facebook network for different teaching and influence probabilities, assuming uniform distribution. 206

A.9 Regret in Facebook network for different teaching and influence probabilities, assuming uniform distribution. 206

A.10 Results for MySpace network across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming uniform distribution. 207

A.11 Results of Influence and Knowledge in MySpace network for different teaching and influence probabilities, assuming uniform distribution. 208

A.12 Regret in MySpace network for different teaching and influence probabilities, assuming uniform distribution. 208

A.13 Results for network A across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming power law distribution. . . 209

A.14 Results of Influence and Knowledge in network A for different teaching and influence probabilities, assuming power law distribution. 210

A.15 Regret in network A for different teaching and influence probabilities, assuming power law distribution. 210

A.16 Results for network B across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming power law distribution. . . 211

A.17 Results of Influence and Knowledge in network B for different teaching and influence probabilities, assuming power law distribution. 212

A.18 Regret in network B for different teaching and influence probabilities, assuming power law distribution. 212

A.19 Results for Facebook network across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming power law distribution. 213

A.20 Results of Influence and Knowledge in Facebook network for different teaching and influence probabilities, assuming power law distribution. . . 214

A.21 Regret in Facebook network for different teaching and influence probabilities, assuming power law distribution. 214

A.22 Results for MySpace network across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming power law distribution. 215

A.23 Results of Influence and Knowledge in MySpace network for different teaching and influence probabilities, assuming power law distribution. 216

A.24 Regret in MySpace network for different teaching and influence probabilities, assuming power law distribution. 216

A.25 Results of 4 real world networks across many interventions, for $p = 0.5$ and $\varphi = 0.5$, assuming power law distribution with $a = 1.2$ 217

A.26 Results of Influence and Knowledge for different teaching and influence probabilities, assuming power law distribution with $a = 1.2$ 218

A.27	Regret for different teaching and influence probabilities, assuming power law distribution with $a = 1.2$. Lower results are better.	218
B.1	Performance metrics over all turns of 9x9 Go games. (Alternative baseline)	220
B.2	ROC curves, analyzing different thresholds in 9x9 Go. (Alternative baseline)	221
B.3	ROC curves for <i>diverse</i> and <i>uniform</i> , for different board sizes. (Alternative baseline)	222
B.4	AUC for different teams and board sizes, organized by teams. (Alternative baseline)	223
B.5	AUC for different teams and board sizes, organized by board sizes. (Alternative baseline)	224
B.6	Differences in prediction quality for the <i>diverse</i> and <i>uniform</i> teams. (Alternative baseline)	225
B.7	Performance metrics over all turns of 9x9 and 21x21 Go games. (Alternative baseline)	226
B.8	Comparison of prediction quality with the full and reduced representation. (Alternative baseline)	227

List Of Tables

3.1	A team of deterministic agents that can reach perfect play under simple voting. “1” indicates agent plays perfect action.	24
3.2	A team of non-deterministic agents that can overcome copies of the best agent.	25
3.3	Probability to select the best move of each player and the teams.	32
3.4	Weak agents can play better in some board states. In parentheses, I show when the difference in P_{best} is 99% significant.	34
3.5	Probability of playing the moves in the first example. * indicates the better move.	39
3.6	Probability of playing the moves in the second example. Some results are unavailable due to lack of memory. * indicates the better move.	40
3.7	Probability of playing the moves in the third example. * indicates the better move.	42
4.1	Performance of diverse team increases when the number of actions increases.	47
4.2	Average winning rates of the team members across different board sizes. Note that these are not the winning rates of the teams.	56
4.3	Winning rates of each one of the agents across different board sizes.	56
5.1	Probability distribution function of the agents in my example.	69
5.2	Probability of outputting each possible optimal solution, for the diverse and uniform teams.	70
5.3	Probability of outputting each possible optimal solution, for different sizes of the diverse team.	71
5.4	GA parameters for the diverse team. Initial Population and Maximum Iteration were kept as constants: 10 and 5, respectively. PZ = Population Size, SZ = Selection Size, CR = Crossover Ratio, MR = Mutation Ratio.	87
6.1	Parameters sampled to generate different versions of Fuego.	102
8.1	A simple example of voting profiles after three iterations of problem-solving.	141
8.2	Example of the full feature vector after three iterations of problem solving.	142
8.3	Example of the reduced feature vector after three iterations of problem solving.	142

Abstract

This thesis introduces a novel paradigm in artificial intelligence: decision-centered teamwork. Decision-centered teamwork is the analysis of agent teams that iteratively take joint decisions into solving complex problems. Although teams of agents have been used to take decisions in many important domains, such as: machine learning, crowdsourcing, forecasting systems, and even board games; a study of a general framework for decision-centered teamwork has never been presented in the literature before.

I divide decision-centered teamwork in three fundamental challenges: (i) **Agent Selection**, which consists of selecting a set of agents from an exponential universe of possible teams; (ii) **Aggregation of Opinions**, which consists of designing methods to aggregate the opinions of different agents into taking joint team decisions; (iii) **Team Assessment**, which consists of designing methods to identify whether a team is failing, allowing a “coordinator” to take remedial procedures.

In this thesis, I handle all these challenges. For **Agent Selection**, I introduce novel models of diversity for teams of voting agents. My models rigorously show that teams made of the best agents are not necessarily optimal, and also clarify in which situations diverse teams should be preferred. In particular, I show that diverse teams get stronger as the number of actions increases, by analyzing how the agents’ probability distribution function over actions changes. This has never been presented before in the ensemble systems literature. I also show that diverse teams have a great applicability for design problems, where the objective is to maximize the number of optimal solutions for human selection, combining for the first time social choice with number theory. All of these theoretical models and predictions are verified in real systems, such as Computer Go and architectural design. In particular, for architectural design I optimize the design of buildings with agent teams not only for cost and project requirements, but also for energy-efficiency, being thus an essential domain for *sustainability*.

Concerning **Aggregation of Opinions**, I evaluate classical ranked voting rules from social choice in Computer Go, only to discover that plurality leads to the best results. This happens because real agents tend to have very noisy rankings. Hence, I create a *ranking by sampling* extraction technique, leading to significantly better results with the Borda voting rule. A similar study is also performed in the social networks domain, in

the context of influence maximization. Additionally, I study a novel problem in social networks: I assume only a subgraph of the network is initially known, and we must spread influence and *learn the graph* simultaneously. I analyze a linear combination of two greedy algorithms, outperforming both of them. This domain has a great potential for *health*, as I run experiments in four real-life social networks from the homeless population of Los Angeles, aiming at spreading HIV prevention information.

Finally, with regards to **Team Assessment**, I develop a *domain independent* team assessment methodology for teams of voting agents. My method is within a *machine learning framework*, and learns a prediction model over the voting patterns of a team, instead of learning over the possible states of the problem. The methodology is tested and verified in Computer Go and Ensemble Learning.

Part I

Introduction and Background

Chapter 1

INTRODUCTION

*They may sound your praise and call you great,
They may single you out for fame,
But you must work with your running mate
Or you'll never win the game;
Oh, never the work of life is done
By the man with a selfish dream,
For the battle is lost or the battle is won
By the spirit of the team.
(Edgar A. Guest)*

Teams of agents have been used for taking decisions while solving complex problems in many important domains. For instance, in machine learning ensemble systems have been widely studied, where the output of multiple classifiers are aggregated into a final classification [Polikar, 2012]. Recently, in the artificial intelligence literature, aggregating the opinions of multiple people has been receiving considerable attention in the study of crowdsourcing [Mao et al., 2013, Bachrach et al., 2012]. Similarly, multiple agents have also been used to make predictions, in the study of forecasting systems [Isa et al., 2010]. Such decision-centered teams have been explored even for board games [Obata et al., 2011, Soejima et al., 2010].

One of the first works to study team-decision making was in 1785, when Condorcet [Condorcet, 1785] analyzed the guarantees of a jury doing majority voting to make a correct verdict, assuming a binary problem (guilt or not guilt) and identical jury members. More recent works in the social sciences literature emphasize the importance of diversity when forming (human) teams, such as Hong and Page [2004], and LiCalzi and Surucu [2012], that study models where agents are able to know the utility of the solutions, and the team can simply pick the best solution found by one of its members.

In machine learning, however, the main focus has been on how to best divide the training set across different learners [Polikar, 2012]. In social choice, the complexity of computing the output of voting rules and of manipulating elections has been widely studied [Bartholdi III et al., 1989a,b, Davies et al., 2011, Yang, 2015], but that still does not guide us on how to form the best voting team. Teamwork models are also a major part of multi-agent systems research, but the focus is on assigning tasks to agents with different capabilities, in order to maximize the expected utility of the team [Nair and Tambe, 2005, Guttman, 2008].

Hence, in this thesis I introduce the concept of *decision-centered teamwork* to study domains where a set of fully cooperative agents is used to take decisions into accomplishing a common goal. Decision-centered teamwork is the analysis of agent teams that iteratively take joint decisions to solve complex problems. This thesis is the first general study of decision-centered teamwork. When forming such teams, we must not only select agents, but also design aggregation mechanisms and domain-independent methods to assess the performance of a given team. Hence, I divide decision-centered teamwork in three fundamental challenges: Agent Selection, Aggregation of Opinions and Team Assessment.

- *Agent Selection* is picking a limited number of agents to form a team. Previous work in social choice considered a single world state [Conitzer and Sandholm, 2005, List and Goodin, 2001], which would lead us to expect the best team to be the one composed by the best possible agents. I show [Marcolino et al., 2013] that this is not true, and it is fundamental to also consider diversity when forming teams. However, my first diversity model only presents necessary conditions for such phenomenon; therefore I present a second, more general model of diversity [Marcolino et al., 2014b], where I can predict that diverse teams perform better than uniform teams (i.e., copies of the best agent) in problems with a large action space. Additionally, I perform a study of diverse and uniform teams for design problems [Marcolino et al., 2016a], where the focus is to maximize the number of optimal solutions for human selection (according to aesthetics or non-formalized compromises).
- *Aggregation of Opinions*, in the context of decision centered teamwork, is combining the opinions of each member of the team into a final team decision. I study ranking extraction and ranked voting rules [Jiang et al., 2014, Yadav et al., 2015] using existing agents in Computer Go and influence maximization. Additionally, I study a linear combination of two greedy algorithms for simultaneously learning and influencing social networks [Marcolino et al., 2016c].

- *Team Assessment* is verifying the performance of a given team. In particular, it is fundamental to predict whether a team is going to be successful or fail in problem solving. Existing methods are tailored for specific domains, such as robot-soccer [Ramos and Ayanegui, 2008]. Hence, I propose [Nagarajan et al., 2015, Marcolino et al., 2016b] a novel *domain independent* technique, which learns a prediction function for estimating the performance of a team using only its voting patterns. Based on such prediction, it is possible to take remedial procedures to increase a team’s performance.

This thesis presents theoretical, methodological and experimental contributions for these facets of decision-centered teamwork. I run experiments in three domains: Computer Go [Marcolino et al., 2013, 2014b, Jiang et al., 2014, Nagarajan et al., 2015, Marcolino et al., 2016b], Architectural Design [Marcolino et al., 2016a] and influence maximization in social networks [Yadav et al., 2015, Marcolino et al., 2016c] (Figure 1.1). As I detail later, these domains have a positive societal impact for *health* and *sustainability*.

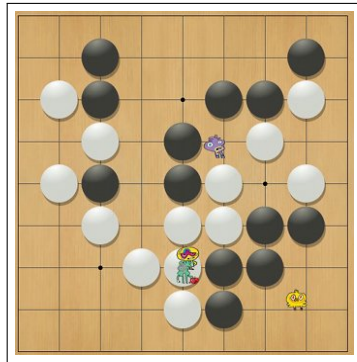
1.1 Summary of Contributions

1.1.1 Agent Selection

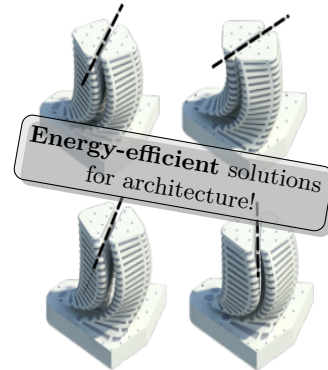
A naïve solution for agent selection would be to simply select the set of the individually best agents to form a team. I show, however, that team diversity is also an important aspect of decision-centered teamwork. Hence, this thesis presents three models of diversity in teamwork.

In my first model, I show that a diverse team can outperform a uniform team if at least one agent has a higher probability of playing the best action than the best agent in at least one world state [Marcolino et al., 2013]. However, since only necessary conditions are provided, I develop a second model [Marcolino et al., 2014b], which study the effect of increasing the number of actions available to choose from. I define *spreading tail* (*ST*) agents, that have an increasingly larger number of actions assigned with a non-zero probability as the number of actions in the domain increases. A diverse team is modeled as a team of *ST* agents. I show that the probability of a diverse team picking the best action increases as the action space increases; and also converges to 1 when the team size grows in large action spaces. The main idea of my model is that diverse agents are less likely to agree on the same mistakes when the action space is large, and therefore only two agents voting for the optimal solution is sufficient. I run experiments in the Computer Go domain, using six existing Computer Go agents, and analyze the predictions and the assumptions of my models.

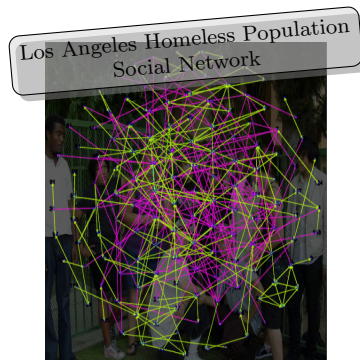
Decision-centered teamwork allowed me to:



Play better Go by voting among multiple diverse agents.



Optimally design buildings by voting among agents and using human selection.



Prevent HIV in uncertain social networks by combining multiple algorithms.

Figure 1.1: Domains where I explored decision-centered teamwork.

My third diversity model focuses on agent teams for design problems. This thesis introduces design problems as a novel problem in social choice, where the performance of a voting system is evaluated by the number of optimal solutions found by an agent team. For maximum applicability, I study agents that are queried for a single opinion, and multiple solutions are obtained by multiple voting iterations. I show that diverse teams composed of agents with different preferences maximize the number of optimal solutions, while uniform teams composed of multiple copies of the best agent are in general suboptimal. I run experiments with diverse and uniform teams in a real system for architectural design optimization. The design of buildings are not only optimized for cost and conformity with project requirements, but also for energy-efficiency. Hence, this is an important domain for sustainability.

1.1.2 Aggregation of Opinions

In the context of decision-centered teamwork, team coordination is aggregating the opinions of multiple agents into taking a final decision. I study the performance of four classical ranked voting rules from social choice in Computer Go. First, I build a ranking for each agent by directly extracting their estimated utilities of each action from their search trees. However, after aggregating such rankings, I find that plurality still had the best winning rates. Such surprising result may seem discouraging at first, but I introduce a novel method to extract rankings from existing agents, based on the frequency that actions are played when sampling an agent multiple times. I show that using this method, the Borda ranked voting rule outperforms plurality. While performing this study, I also introduce a technique for generating diverse teams by creating random parametrizations of a *base* agent [Jiang et al., 2014], in order to analyze larger diverse teams of voting agents. Similarly, I analyze the performance of ranked voting aggregation in the social networks domain, in the context of influence maximization [Yadav et al., 2015].

Additionally, I analyze a linear combination of two greedy algorithms in a novel social networks problem: I assume only a subgraph of the network is initially known, and we must spread influence and *learn the graph* simultaneously [Marcolino et al., 2016c]. Although influence maximization has been extensively studied [Kempe et al., 2003, Cohen et al., 2014, Golovin and Krause, 2010, Maghami and Sukthankar, 2012, Li and Jiang, 2014], the main motivation of previous works is viral marketing, and hence they assume that the social network graph is fully known beforehand, generally taken from some social media network (such as Facebook or MySpace). However, the graphs recorded in social media do not really represent all the people and all the connections of a population. Most critically, when performing interventions in real life, we deal with large degrees of lack of knowledge.

I show that the traditional greedy influence maximization algorithm may have arbitrarily low performance. However, we can still effectively influence and learn a network graph when the *independence of objectives* hypothesis holds (which will be later defined in this thesis). When it does not hold, I give an upper bound for the influence loss, which converges to 0 as the number of interventions in the network grows. I run extensive experiments over four real-life social networks of the homeless population of Los Angeles.

Influence maximization has a great potential to positively impact society, as we can modify the behavior of a community. For example, we can increase the overall health of a population; my main motivation is to spread information about HIV prevention in the homeless population of Los Angeles.

1.1.3 Team Assessment

Team assessment is essential for decision-centered teamwork. We must be able to assess if a team is likely to succeed or fail in order to take remedial procedures when appropriate. For instance, we could switch team members or the aggregation mechanisms according to the current assessment, in order to increase the team performance in an online manner. Hence, it is important to develop methods to quickly assess a team’s performance.

I propose a model for team assessment [Nagarajan et al., 2015, Marcolino et al., 2016b], which allows me to develop a novel domain independent technique to predict online the final reward of a team. The final reward is defined by a random variable; which is influenced by a set of variables representing the subset of agents that agreed on the chosen action at each world state. Based on that, I show that the final reward can be predicted by linear models, and I derive a domain-independent prediction function using only the frequencies of agreement of each subset of agents. Additionally, I show that the quality of the prediction increases as the action space grows larger.

I evaluate the quality of the predictions using a real system of Computer Go playing agents, over a varying of board sizes. I study the predictions at every turn of the games, and compare with an analysis performed by using an in-depth search. I achieve an accuracy of 71% for a diverse team in 9×9 Go, and of 81% when I increase the action space size to 21×21 Go. For a uniform team, I obtain 62% accuracy in 9×9 , and 75% accuracy in 21×21 Go. I also predict the final performance of an ensemble system that votes to classify hand-written digits, demonstrating the domain independence of my method.

1.2 Guide to thesis

This thesis is organized as follows: Chapter 2 introduces the related work and the necessary background for the research presented in this thesis. Part II introduces the work on agent selection: Chapter 3 presents my first diversity model; Chapter 4 shows my second model, where diverse teams in large action spaces are analyzed; and Chapter 5 presents my model about agent teams for design problems. Part III introduces my work on aggregation of opinions: Chapter 6 presents an experimental study on ranked voting rules in Computer Go and Social Networks, and Chapter 7 studies a linear combination of two algorithms for simultaneously influencing and mapping social networks. Part IV focuses on team assessment, where Chapter 8 introduces a domain independent technique to predict success or failure in teams of voting agents. Finally, Chapter 9 presents my conclusions and discussions for future work.

Chapter 2

BACKGROUND AND RELATED WORK

If I have seen further it is by standing on the shoulders of Giants.
(Isaac Newton)

In this chapter I discuss relevant background and related work for this thesis. The study of decision-centered teamwork relates to different areas of study in Computer Science, and even in other fields, such as Social Science. Hence, in this section I will discuss the literature in Multi-agent Systems, focusing on the study of Social Choice (Section 2.1) and Teamwork (Section 2.2); Machine Learning, especially Ensemble Systems (Section 2.4); and Social Science (Section 2.3), where many models have been proposed to study diversity in human teams. Since the thesis also introduces the problem of simultaneously influencing and mapping social networks (in Chapter 7), I will also discuss the relevant literature for this problem in Section 2.7.

2.1 Social Choice

Social choice studies theoretical frameworks for analyzing the combination of individual preferences or opinions into a collective decision. One of the most common methods to aggregate the opinions of different agents into taking a decision is voting. Hence, voting is extensively studied in the social choice literature. Normally, voting is seen under one of two possible views: (i) as a way to decide on a fair outcome, given the preferences of different individuals; (ii) as a way to use the opinions of different agents to estimate a correct outcome, or a “truth”.

The classical work in the second view is the Condorcet’s Jury Theorem [Condorcet, 1785], published in the 18th century. Condorcet developed the theorem to justify the use of voting in the jury system. According to his theorem, when facing a binary decision, as long as the average of the probability of each individual being correct is higher than $\frac{1}{2}$, a

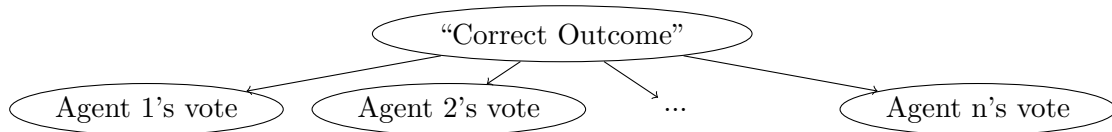


Figure 2.1: “Classical” voting model. Each agent has a noisy perception of the truth, or correct outcome. Hence, its vote is influenced by the correct outcome.

group of independent individuals doing majority voting will have a higher probability of being correct than the individuals alone. In fact, as the number of voters go to infinite, the probability of taking a correct decision converges to 1. This theorem is extended to the k options case in List and Goodin [2001], where it is shown that if each of the individuals have a probability of choosing the best answer higher than choosing any other answer, the group performing majority voting will still be stronger than the individuals alone.

Recently, social choice has been considered extensively in the artificial intelligence literature. Each agent is modeled as having a noisy perception of the truth (or correct decision) [Conitzer and Sandholm, 2005]. Hence, the correct outcome influences how each agent is going to vote, as shown in the model in Figure 2.1.

Therefore, given a voting profile and a noise model (the probability of voting for each action, given the correct outcome) of each agent, we can estimate the likelihood of each action being the best by a simple (albeit computationally expensive) probabilistic inference. Any voting rule is going to be optimal if it corresponds to always picking the action that has the maximum likelihood of being correct (i.e., the action with maximum likelihood of being the best action), according to the assumed noise model of the agents. That is, the output of an optimal voting rule always corresponds to the output of actually computing, by the probabilistic inference method mentioned above, which action has the highest likelihood of being the best one.

Classically, however, this view of voting considers only teams of identical agents [List and Goodin, 2001, Conitzer and Sandholm, 2005], and thus they do not consider the importance of diversity when selecting agents. More recent works are also considering agents with different probability distribution functions. Caragiannis et al. [2013] study which voting rules converge to a true ranking as the number of agents (not necessarily identical) goes to infinity. In Soufiani et al. [2012] the problem of inferring the true ranking is studied, assuming agents with different pdfs, but drawn from the same family. Therefore, team formation is not the focus of their work.

Since the classical works of Bartholdi III et al. [1989a,b], many works in social choice also study the computational complexity of computing the winner in elections [Aziz et al., 2015, Amanatidis et al., 2015], and/or of manipulating the outcome of an election, in

general by disguising an agent’s true preference [Conitzer and Sandholm, 2003, Davies et al., 2011, Yang, 2015]. The main motivation of such works is in finding voting rules which are easy to compute (i.e., the winner can be computed in polynomial time), but are hard to manipulate (i.e., an agent would have to run an NP-Complete algorithm in order to calculate a voting profile different than its true preference in order to change the final outcome of the voting procedure).

There are also works studying the aggregation of partial [Xia and Conitzer, 2011a,b] or non-linear rankings (such as pair-wise comparisons among alternatives) [Elkind and Shah, 2014], since it could be costly/impossible to request agents for a full linear ranking over all possible actions. Some very recent works in social choice also analyze probabilistic voting rules, where the agents’ votes affect a probability distribution over outcomes [Brandl et al., 2015, Chatterji et al., 2014].

2.1.1 Large Solutions Pool

In this thesis I also consider agent teams for design problems, where the objective is to maximize the number of optimal solutions. In Chapter 5 I discuss in detail why for design problems it is important to obtain a large number of solutions. In this chapter I present the current state of the art in social choice with regards to obtaining a large solution pool.

Most of the social choice literature is about finding a correct ranking in domains where there is a linear order over the alternatives, and hence a unique optimal decision [Conitzer and Sandholm, 2005, Caragiannis et al., 2013, List and Goodin, 2001, Soufiani et al., 2012, Baharad et al., 2011]. Recent works, however, are considering more complex domains. Farfel and Conitzer [2011] analyze the case where instead of having a single preferred action, voters have a preference over an interval of real numbers. Xia and Conitzer [2011a] study the problem of finding k optimal solutions, where k is known beforehand, by aggregating rankings from each agent. However, not only do they need strong assumptions about the quality of the rankings of such agents, but they also show that calculating the *maximum likelihood estimation* (MLE) from the rankings is an NP-hard problem.

Procaccia et al. [2012] study a similar perspective, where the objective is to find the top k options given rankings from each agent, where, again, k is known in advance. However, in their case, they assume there still exists one unique truly optimal choice, hidden among these top k alternatives. Elkind and Shah [2014], motivated by the crowdsourcing domain, study the case where instead of rankings, the voters output pairwise comparisons among all actions, which may not follow transitivity. However, their final objective is still to pick a single winner, not to maximize the set of optimal solutions found by a voting

system. Oren and Lucier [2014] study the problem of finding the top k options in an on-line setting, where we must take a decision concerning adding a candidate to a k -sized pool at each time we see the (ranked) vote of an agent; hence the problem of maximizing the number of optimal solutions is not considered.

Finally, outputting a full comparison among all actions can be a burden for an agent [Boutilier, 2002, Kalech et al., 2011]. In this thesis I show that actual agents can have very noisy rankings, and therefore do not follow the assumptions of previous works in social choice. Hence, as any agent is able to output at least one action (i.e., a single vote), I study here systems where agents vote across multiple iterations, which brings the model closer to real applications for design.

2.2 Teamwork

Teamwork is an important topic of research in the multi-agent systems literature. Many theoretical frameworks of teamwork have been developed. For example, Cohen and Levesque [1991] introduced the Joint Intentions theory. In that framework, a team has a joint mental state. All agents work to achieve a certain objective in the joint mental state. If one of the agents discovers that the objective has been achieved, or became irrelevant/impossible, then it must communicate with its teammates in order to pass this belief to the joint mental state. In the SharedPlans [Grosz and Kraus, 1996] framework, there is a set of possible recipes for achieving one action, which are composed by subactions, forming a hierarchy; and agents may have individual plans to perform some of the subactions. These ideas are combined in an actual implemented framework in STEAM [Tambe, 1997], where agents build a hierarchy of joint intentions when performing tasks in three different domains. STEAM is further extended in Tambe et al. [2002], where a markov decision process (MDP) model is proposed, enabling agents to autonomously decide when to transfer control (i.e., decision-making) to humans or other agents.

A common approach to coordination is to consider a centralized controller, that plans beforehand the best action each agent should take at each world state (i.e., plans *offline*); and the agents merely execute the actions ascribed to them in the plan. These are normally studied in the Dec-POMDP framework (decentralized partially observable Markov decision process) [Bernstein et al., 2002], and policy iteration is one of the main methods to compute the global plan (i.e., a policy) [Bernstein et al., 2009].

Task allocation is also an important approach to coordinate a team. A problem or goal is distributed in a set of tasks, and each agent must execute a subset of those. The contract net protocol [Smith, 1980] is a very common technique for task allocation, where agents can be manager and contractors. A manager receives bids and allocate a task to

the most appropriate agent. Upon being allocated a task, the agent (contractor) must execute it, but it can divide those in subtasks and also act as a manager to allocate those. A similar approach is the auction based task allocation mechanism [Bertsekas, 1992], where agents submit bids to compete for tasks, like in actual auctions.

For large teams, especially swarms, a common approach is to specify individual behaviors for each agent, in order for a desired global behavior to emerge in a distributed fashion [Marcolino and Chaimowicz, 2008, 2009a,b]. Designing the individual behaviors in an automatic way, however, is still an open challenge.

Decision-centered teamwork is a fundamentally different form of teamwork from the ones previously described. Instead of each agent executing one action, or solving a certain subtask of a common goal, the team must decide on a single team action at each world state towards problem solving. However, it is still a form of teamwork, as all agents are fully cooperative, and they share a common goal. Aggregating their opinions at each step of problem-solving is their method of achieving that common goal.

2.2.1 Team Formation

Team formation is also an active topic of research in multi-agent teamwork. However, the main focus has been in modeling tasks as requiring a set of skills; and the team formation problem is considered as selecting a set of agents with all the necessary skills but the minimum cost [He and Ioerger, 2003, Guttman, 2008]. More recent work go beyond a simple sum of skills and also models the synergy of a group [Liemhetcharat and Veloso, 2012], or how to automatically configure a network of agents [Gaston and desJardins, 2005]. In Matthews et al. [2012], a team formation procedure is presented for a class of online football prediction games, and the system is able to play successfully against a large number of human players.

Recently, ad-hoc teamwork is also surging as an important area of study in the multi-agent literature [Stone et al., 2013], where agents must coordinate without being pre-programmed to work together. Researchers have considered, for example, how to lead a group to the optimal joint action with a new ad-hoc agent [Agmon and Stone, 2012], or how to influence a flock of agents [Genter and Stone, 2014]. Barret and Stone [2015] have demonstrated ad-hoc teamwork in practice in the simulated robot soccer domain. These works, however, do not analyze the importance of diversity when forming agent teams.

2.3 Social Science

Forming teams to take decisions has also been studied in the social sciences. For example, Hong and Page [2004] is an impactful work showing the importance of diversity when

forming (human) teams. Even though recently some of the mathematical arguments were put into question [Thompson, 2014], it remains as a mile-stone on the study of the importance of diversity, as many researchers were influenced by their work [Luan et al., 2012, Lakhani et al., 2007, Krause et al., 2011], showing the importance of diversity in different settings. In their model, each agent has a set of local minima that they reach while trying to maximize an objective function. The agents can improve the solution from the local minima of their team members, therefore the search of a team stops only in the intersection of the local minima of all agents. By using a large number of diverse agents the system is able to converge to the optimal solution. Their model, however, does not cover situations where agents are unable to improve the solution from their team members' local minima. This can happen, for example, when we use existing software, that were not architected to collaborate in this way or when there are time constraints. Therefore, there are many situations where the agents have to collaborate in other ways, such as voting. If a team of agents votes, the system will not necessarily converge to an option in the intersection of their local minima. However, as I will show, it is still possible for a diverse team to play better than a uniform strong team.

A more recent model to analyze diversity was proposed in LiCalzi and Surucu [2012]. It is an equivalent model to Page's and still do not overcome the limitations previously described. In Braouezec [2010], the authors show the benefits of diverse agents voting to estimate the optimum of a single peaked function. In this thesis I am dealing with a harder problem, as the function to be optimized changes at every iteration. Another work that uses voting to study diversity is West and Dellana [2009], but they assumed that Page's model would work in a voting context, and do not propose a new model. Lamberson and Page [2012] study diversity in the context of forecasts. They assume that solutions are represented by real numbers, and a team converges to the average of the opinion of its members. Hence, they do not capture domains with discrete solutions, and the model also does not cover teams of voting agents.

Prediction markets are also studied in the social sciences. They are markets where the users bet on possible outcomes of real world events. For example, if a user u believes that an event x will take place with probability p_u (for instance, a certain candidate winning an election), then she can offer to buy a share on x with a price p_u . Similarly, a user can also offer to buy a share that an event x will not take place, and the price p_u will correspond to the probability of the event not taking place. The final market price of the event x can be interpreted as the probability of x actually taking place in the world. Erikson and Wlezien [2008], for example, compare the accuracy of prediction markets in determining the final outcome of elections with pools. Manski [2006] presents the first theoretical study concerning to what extent the market prices can be actually understood as the

probabilities of events occurring. A historical analysis of political prediction markets in the USA is presented at Rhode and Strumpf [2004]. In this thesis, however, I will focus on voting as a way to aggregate opinions, rather than markets.

2.4 Machine Learning

Combining multiple classifiers through voting is a very common technique in machine learning [Polikar, 2012]. Such systems are called Ensemble Systems. Traditionally, each classifier runs the same algorithm, but trained with different subsets of items, or different features. The way that each subset of items is selected for each classifier defines different ensemble system approaches, for example: bagging, boosting, or ada-boost; while mixture of experts trains each classifier in a different subspace of the feature space.

Combining multiple classifiers has been a very active research area. For example, Sylvester and Chawla [2005] uses a Genetic Algorithm to learn an optimal set of weights when combining multiple classifiers, Chiu and Webb [1998] uses voting to predict the future actions of an agent, and AL-Malaise et al. [2014] use ensembles to predict the performance of a student.

Diversity is known to be important when forming an ensemble, and some systems try to minimize the correlation between the classifiers while training [Chen and Yao, 2009]. Still, an important problem is how to form the ensemble system, i.e., how to pick the classifiers that lead to the best predictions [Fu et al., 2012].

My models allow us to make many predictions about teams as the action space and/or number of agents changes, and also compare the rate of change of the performance of different teams, or analyze agent teams in the context of design problems. To the best of my knowledge, there are no models similar to mine in the machine learning literature.

Additionally, team assessment is also important for ensembles, as we need techniques to evaluate whether the current ensemble is performing well in order to change it if necessary. In Chapter 8 I will perform experiments in predicting the final performance of an ensemble system.

2.5 Team Assessment

As in Chapter 8 I discuss a technique for assessing the performance of a team of voting agents, in this section I discuss relevant literature in team assessment.

Traditional team assessment methods rely heavily on tailoring for specific domains. Raines et al. [2000] present a method to build automated assistants for post-hoc, offline team analysis; but domain knowledge is necessary for such assistants. Other methods

for team analysis are heavily tailored for robot-soccer, such as Ramos and Ayanegui [2008], that present a method to identify the tactical formation of soccer teams (number of defenders, midfielders, and forwards). Mirchevska et al. [2014] present a domain independent approach, but they are still focused on identifying opponent tactics, not in assessing the current performance of a team.

In the multi-agent systems community, we can see many recent works that study how to identify agents that present faulty behavior [Khalastchi et al., 2014, Lindner and Agmon, 2014, Tarapore et al., 2013]. Other works focus on verifying correct agent implementation [Doan et al., 2014] or monitoring the violation of norms in an agent system [Bulling et al., 2013]. Some works go beyond the agent-level and verify if the system as a whole conforms to a certain specification [Kouvaros and Lomuscio, 2013], or verify properties of an agent system [Hunter et al., 2013]. However, a team can still have a poor performance and fail in solving a problem, even when the individual agents are correctly implemented, no agent presents faulty behavior, and the system as a whole conforms to all specifications.

Sometimes even correct agents might fail to solve a task, especially embodied agents (robots) that could suffer sensing or actuating problems. Kaminka and Tambe [1998] present a method to detect clear failures in an agent team by social comparison (i.e., each agent compares its state with its peers). Such an approach is fundamentally different than this work, as we are detecting a tendency towards failure for a team of voting agents (caused, for example, by simple lack of ability, or processing power, to solve the problem), not a clearly problematic situation that could be caused by imprecision/failure of the sensors or actuators of an agent/robot. Later, Kaminka [2006], Kalech and Kaminka [2007], Kalech et al. [2011] study the detection of failures by identifying disagreement among the agents. In my case, however, disagreements are inherent in the voting process. They are easy to detect but they do not necessarily mean that a team is immediately failing, or that an agent presents faulty behavior/perception of the current state.

There is also a body of work that focuses on analyzing (and predicting the performance of) *human* teams playing sports games. For example, Quenzel and Shea [2014] learn a prediction model for tied NFL American football games, where they use logistic regression to predict the final winner. They study the coefficients of the regression model to determine which factors affect the final outcome with statistical significance. Heiny and Blevins [2011] use discriminant analysis to predict which strategy an American football team will adopt during the game. In soccer, Bialkowski et al. [2014] analyze data from games to automatically identify the roles of each player, and Lucey et al. [2015] also use logistic regression to predict the likelihood of a shot scoring a goal. We can also find examples in basketball. Maheswaran et al. [2012] use a logistic regression model

to predict which team will be able to capture the ball in a rebound, while Lucey et al. [2014] study which factors are important when predicting whether a team will be able to perform an open 3-points shot or not.

2.6 Other Multiple Algorithms Techniques

There is a vast literature on using multiple algorithms to solve complex problems. For example, in algorithm portfolios [Gomes and Selman, 2001], multiple programs run in parallel to solve a single problem instance, as one of them may be the fastest for the particular instance. In multi-start local search algorithms [Gyorgy and Kocsis, 2011], many searches run in parallel, and we must dynamically decide to allocate computational resources to a search process or to initialize a new search. There are also works about multiple experts systems, where predictions are performed by choosing one among multiple experts' advice [Cesa-Bianchi and Lugosi, 2006]. In the robotics literature, multi-heuristic A* search has recently been proposed [Aine et al., 2014], where multiple heuristics are used simultaneously to find a bounded suboptimal solution.

Concerning distributed optimization, Chapter 5 is related to the study of distributed genetic algorithms [Knysh and Kureichik, 2010]. The experimental section on agent team for designs (Section 5.5) relates to the “island model”, where populations evolve concurrently. This model has been improved recently, by Osaba et al. [2015], with a technique where some of the populations stop evolving temporarily in order to focus the search on promising ones. Normally, however, the populations interact by transferring offsprings, not by voting, and a theoretical study of voting teams which must maximize the number of optimal solutions was never performed. Other recent works study how to run genetic algorithms on GPUs in order to solve classical NP-hard problems or navigate UAVs [Cekmez et al., 2013, 2014].

2.7 Social Networks

As in Chapter 7 I will focus on the problem of influencing social networks that are not fully known in advance, I discuss here related works in influence maximization.

The influence maximization problem has recently been a very popular topic of research. Normally, the main motivation is viral marketing in social media (like Facebook or MySpace). Hence, previous works assume full knowledge of the social network graph. The classical result is Kempe et al. [2003], where they study the approximation guarantee of a greedy algorithm for the “independent cascade model”, which will be described in Chapter 7. Golovin and Krause [2010] extended that result to the case where we are able

to observe which nodes are already influenced or not before picking each one. However, they still assume full knowledge of the graph. Similarly, Dhamal et al. [2015] study the specific situation where we select two subsets of nodes at two different phases, and we can observe the network between the phases. Cohen et al. [2014] focus on how to quickly estimate the potential spread of one node, since running simulations (as needed by the greedy algorithm) in large graphs is computationally expensive. A different view was studied by Yadav et al. [2015], where they analyze a model where nodes try to influence their neighbors multiple times, and each edge has an existence probability. Here, however, I will deal with a different kind of uncertainty, as in my model whole portions of the graph are completely unknown.

Some works in the AAMAS (multi-agent systems) community study variations of the original influence model. For example, Pasumarthi et al. [2015] study the problem where the objective is to maximize influence not in the whole social network graph, but only in a subset of nodes which represent targeted consumers of a certain product. Anagnostopoulos et al. [2015] study the case where multiple entities compete to spread influence in order to convince consumers to buy their respective products. Maghami and Sukthankar [2012] consider influence maximization under more realistic assumptions than traditional models: they assume agents pertain to social groups, and the ones of the same group are more likely to influence each other. Moreover, they assume multiple products, and that each agent has a certain probability of buying a particular one. Li and Jiang [2014] study a different way to increase the realism: they consider graphs with multiple edge types, modeling different connection types between people (for example, Facebook or real life interaction). Tsang and Larson [2014], instead of focusing on the problem of selecting subset of nodes, run several experiments to study the evolution of social networks, assuming that each node has a degree of adoption (instead of a binary value representing influenced or not influenced), and nodes are more likely to be influenced by others with a similar adoption degree. However, even though all these works are more realistic than the traditional models, they still assume full knowledge of the social network graph.

Chapter 7 is also related to the problem of submodular optimization (as influence is a submodular function) by selecting elements without knowledge of the whole set. Badanidiyuru et al. [2014] present an algorithm for selecting the best subset, giving elements arriving from a “stream”. Hence, in their case, given enough time the whole set would be seen, and which elements are discovered does not depend on which ones are selected.

My problem is also related to the classical max- k -cover problem [Khuller et al., 1999], where we must pick k subsets that maximize our coverage of a set. In that case, however,

the elements of each subset is known. There is also an online version of the max-k-cover problem [Alon et al., 2009], where we pick one subset at a time, but an adversary fixes which element must be covered next. However, the set and the available subsets are still known in advance. Similarly, Grandoni et al. [2008] study the case of covering a set whose elements are randomly chosen from a universe of possible elements. Another important related problem is the submodular secretary problem [Bateni et al., 2013], where we again pick a subset to optimize a submodular function. However, in that case, we receive one element at a time, and we must make an irrevocable decision of either keeping it or not.

The problem of learning a graph online was studied in the context of security games by Barth et al. [2010], where there is a possible attack graph with unknown edges. They learn edges when they are used by an attacker for the first time, and repeatedly update the probability of protecting each edge using an online learning algorithm [Freund and Schapire, 1999]. In my case, however, we cannot update weights iteratively, as we pick a node at most one time.

Finally, Chapter 7 also relates to sequential decision making with multiple objectives [Roijsers et al., 2013]. Here I do not aim at computing an optimal policy (which is computationally expensive), but at studying a greedy method, similar to other works in influence maximization. My algorithm is a scalarization over two objectives, but such method was never studied before to influence and map social networks.

Part II

Agent Selection

Chapter 3

DIVERSITY BEATS STRENGTH?

*I exclude no one I am strengthened by all
My name is Diversity and yes I stand tall.
Recognize me and keep me in the mix
Together there's no problem that we can't fix.
I am your best hope towards true innovation
And to many, I reflect hope and inspiration.
(Charles Bennafield)*

3.1 Introduction

Team formation is essential when dealing with a multi-agent system. Given limited resources, we must select a strong team to deal with a complex problem. Many works model team formation as selecting a team that accomplishes a certain task with the maximum expected value, given a model of the capabilities of each agent [Nair and Tambe, 2005, Guttman, 2008]. Other works go beyond a simple sum of skills, for example by considering synergetic effects in a team of agents [Liemhetcharat and Veloso, 2012] or studying how to automatically configure a network of agents [Gaston and desJardins, 2005].

After forming a team, their members must work together. There are many different ways for a team to coordinate. One common and simple way is to use voting. By voting, a team of agents can get closer to finding the best possible decision in a given situation [List and Goodin, 2001]. Only one voting iteration might not be enough, sometimes the agents must vote continuously in many different scenarios. Consider, for example, agents that are cooperating in a board game [Obata et al., 2011, Soejima et al., 2010], deciding together stock purchases across different economic scenarios, or even picking

items to recommend to a large number of users [Burke, 2002]. This situation imposes a conflict for team formation: should we focus on the diversity of the team or on the strength of each individual member? Previous works do not address this issue. Diversity is proposed as an important concept for team formation in the field of Economics and Social Science [Hong and Page, 2004, LiCalzi and Surucu, 2012]. However, Hong and Page [2004], LiCalzi and Surucu [2012] assume a model where each agent brings more information, and the system converges to one of the best options known by the group. When a team votes to decide its final opinion, their model and theorems do not hold anymore. In the current literature on voting it is assumed a model where agents have a fixed probability to take the best action [Condorcet, 1785, List and Goodin, 2001, Young, 1995, Conitzer and Sandholm, 2005, Xia, 2011], and under that model it is not possible to show any advantage in having a diverse team of agents. My experiments show, however, that a diverse team can outperform a uniform team of stronger agents. It is necessary to develop, therefore, a new model to analyze a team of voting agents.

In this chapter, I present a new model of diversity and strength for a team of voting agents. The fundamental novelty of my model is to consider a setting with multiple world states, and each agent having different performance levels (characterized by different probability distributions) across world states. Under this model, I can show that a team of diverse agents can perform better than a uniform team composed by strong agents. I present the necessary conditions for a diverse team to play better than a uniform team, and study optimal voting rules for a diverse team. I show synthetic experiments with a large number of teams that demonstrate that both diversity and strength are important to the performance of a team. I also show results in one of the main challenges for Artificial Intelligence: Computer Go. Go is an iterative game, and the possible board states can represent a great variety of different situations, in such a way that the relative strength of different Go playing software changes according to the board state. Therefore, we can use my model to study a team of agents voting to play Computer Go. By using a diverse team I am able to increase the winning rate against Fuego (one of the strongest Go software) by 18.7%, and the diverse team could play 11% better than a team of copies of Fuego. Moreover, the diverse team plays 15.8% better than one of the versions of parallelized Fuego. I also improve the performance of the diverse team by 12.7% using one of my proposed voting rules. Therefore, I effectively show that a team of diverse agents can have competitive strength, and even play better, than a uniform team composed by stronger agents. My new model provides a theoretical explanation for my results.

3.2 Methodology

Let Φ be a set of agents ϕ_i voting to decide an action a in the set of possible actions \mathbf{A} and Ω be a set of world states ω_j . I assume that we can rank the actions from best to worst and \mathbf{U}_j is the vector of expected utilities of the actions in world state ω_j , ordered by rank. The agents do not know the ranking of the actions, and will vote according to some decision procedure, characterized by a probability distribution function (pdf) over action ranks. Hence, each agent ϕ_i has a pdf $\mathbf{V}_{i,j}$ for deciding which action to vote for in state ω_j . Agents that have the same $\mathbf{V}_{i,j}$ in all world states will be referred as copies of the same agent.

Let α_j be the likelihood of world state ω_j . If we expect the world states to be equally frequent, we can use $\alpha_j = 1/|\Omega|$. I define strength as the weighted average of the expected utility of an agent or a team. It is given by the following dot product: $s = \sum_{\omega_j \in \Omega} \alpha_j \mathbf{V}_j \cdot \mathbf{U}_j$, where \mathbf{V}_j is the pdf of the agent/team in world state ω_j . \mathbf{V}_j can be calculated given a team of agents and a voting rule. A voting rule is a function that given the (single) votes of a team of agents, outputs an action.

I define the team formation problem as selecting from the space of all possible agents Ψ a set of n agents Φ that has the maximum strength in the set of world states Ω . An application does not necessarily know $\mathbf{V}_{i,j}$ for all agents and for all world states. In this chapter, I focus on showing that the naïve solution of forming a team by selecting the strongest agents (or copies of the best agent) is not necessarily the optimal solution. Therefore, I am introducing a new problem to the study of team formation.

I define diversity as how different are the probability distributions of agents in Φ in the set of world states Ω : $d = \frac{1}{|\Phi|^2} \sum_{\omega_j \in \Omega} \sum_{\phi_i \in \Phi} \sum_{\phi_k \in \Phi} \alpha_j H(\mathbf{V}_{i,j}, \mathbf{V}_{k,j})$, where H is a distance measure between two pdfs. In this chapter, I use the Hellinger Distance [Hellinger, 1909], given by: $H(\mathbf{V}_{i,j}, \mathbf{V}_{k,j}) = \frac{1}{\sqrt{2}} \sqrt{\sum_{a \in \mathbf{A}} (\sqrt{V_{i,j}(a)} - \sqrt{V_{k,j}(a)})^2}$.

At each iteration, each agent will examine the current world state and submit its (single) opinion about which one should be the next action. The opinions are then combined using plurality voting, that picks as a winner the option that received the most votes. I consider in this chapter three different voting rules: *simple* - break ties randomly, *static* - break ties in favor of the strongest agent overall, *optimal* - break ties in favor of the strongest agent of each world state. I consider the *static* voting rule because in some applications we might have a clear idea of which is the strongest agent overall, but the information of which is the strongest agent for a given world state might not be available. I will encounter this situation in the Computer Go domain, as will be clear in Section 3.3.2. This voting procedure will repeat at every iteration, until the end, when the system can obtain a reward.

Agent	State 1	State 2	State 3	State 4	Strength
Agent 1	1	0	1	1	0.75
Agent 2	0	1	1	0	0.5
Agent 3	1	1	0	0	0.5
Agent 4	1	1	0	1	0.75
Agent 5	0	0	1	1	0.5

Table 3.1: A team of deterministic agents that can reach perfect play under simple voting. “1” indicates agent plays perfect action.

3.2.1 Diversity Beats Strength

I first present examples to demonstrate that a diverse team can play better than a uniform team. First, let’s consider the simplest case, when all agents are deterministic. The team made of copies of the strongest agent will play as well as the strongest agent, no matter how many members we add in the team. However, a team of diverse agents can overcome the strongest agent, and even reach perfect play, as we increase the number of agents. Consider, for example, the team in Table 3.1. This diverse team of 5 agents will reach perfect play under simple voting, while copies of the best agent (Agent 1 or Agent 4) will be able to play well only in 3 out of 4 world states, no matter how many agents we use in the team.

We can easily change the example to non-deterministic agents, by decreasing slightly the probability of them playing their deterministic action. An example is shown in Table 3.2, where I show the pdf of the agents for each world state. I considered the utility vector $\langle 1, 0, 0 \rangle$ for all world states. The resulting strength of the teams is very similar to the deterministic case. Assuming all world states are equally likely, the strength of the diverse team is 0.9907, while copies of the best agent have strength 0.7499. Therefore, it is possible for a team of weak diverse agents to overcome a uniform team of stronger agents, when in certain states the individual agents are stronger than the overall strongest agent.

Even if we make the number of agents go to infinity, copies of the best agent will still be unable to perform the best action in one world state, and will play worse than the diverse team with only five agents. This situation is not considered in the Condorcet’s Jury Theorem, neither in the classical nor in the extended version, because they assume independent agents with a fixed pdf. Therefore, in the previous models, we would not be able to show the importance of diversity.

Agent	State 1	State 2	State 3	State 4
Agent 1	$\langle 0.99, 0.01, 0 \rangle$	$\langle 0, 0.99, 0.01 \rangle$	$\langle 0.99, 0, 0.01 \rangle$	$\langle 0.99, 0.01, 0 \rangle$
Agent 2	$\langle 0, 0.99, 0.01 \rangle$	$\langle 0.99, 0.01, 0 \rangle$	$\langle 0.99, 0, 0.01 \rangle$	$\langle 0, 0.01, 0.99 \rangle$
Agent 3	$\langle 0.99, 0.005, 0.005 \rangle$	$\langle 0.99, 0.005, 0.005 \rangle$	$\langle 0, 0.5, 0.5 \rangle$	$\langle 0, 0.5, 0.5 \rangle$
Agent 4	$\langle 0.99, 0.01, 0 \rangle$	$\langle 0.99, 0.004, 0.006 \rangle$	$\langle 0, 0.4, 0.6 \rangle$	$\langle 0.99, 0.003, 0.007 \rangle$
Agent 5	$\langle 0, 0.3, 0.7 \rangle$	$\langle 0, 0.7, 0.3 \rangle$	$\langle 0.99, 0.005, 0.005 \rangle$	$\langle 0.99, 0.002, 0.008 \rangle$

(a) Agents' pdfs

Agent	Strength
Agent 1	0.7425
Agent 2	0.4950
Agent 3	0.4950
Agent 4	0.7425
Agent 5	0.4950

(b) Agents' strength

Table 3.2: A team of non-deterministic agents that can overcome copies of the best agent.

3.2.1.1 Necessary Conditions

I present a formal proof of the conditions necessary for a diverse team to play better than copies of the best agent, under the simple voting rule. If the conditions of the theorem are not met, we can simply use copies of the best agent as the optimal team. To simplify the presentation of the proof, we will consider a utility function with a value of 1 for the optimal action and 0 for the other actions. That is, we will consider the optimal team in a fixed world state as the team that has the highest probability of performing the optimal action. Let ψ_{best} be the strongest agent in Ψ , and a_{best} be the best action in a given world state.

Theorem 3.2.1 *For a diverse team to be the optimal team under the simple voting rule it is necessary that at least one agent in Ψ has a higher probability of taking the best action than ψ_{best} in at least one world state, or a lower probability of taking a suboptimal action than ψ_{best} in at least one world state.*

Proof: I develop the proof by showing that copies of the best agent of a given world state will be the optimal team in that world state. Therefore, it is necessary that the agents in the diverse team play better than the best agent overall in at least one world state. Let $\psi_{best,j}$ be the strongest agent in world state ω_j . Let's define the pdf of this agent as $\langle p_1, \dots, p_k \rangle$, where p_1 is the probability of taking the best action. I will show that a team of n copies of $\psi_{best,j}$ doing simple voting will have a higher probability of taking the best action than a team of n agents composed of x copies of $\psi_{best,j}$ and m agents ψ_i doing simple voting, where the probabilities of each ψ_i are given by $\langle p_1 - \epsilon_i, p_2 + \gamma_{i,2}, \dots, p_k + \gamma_{i,k} \rangle$, $\gamma_{i,l} \geq 0 \forall l \in (2, k)$ and $\sum_{l=2}^k \gamma_{i,l} = \epsilon_i$.

Given a team of agents, let them all vote. We will start with a team of x copies of agent $\psi_{best,j}$. We will perform m iterations, and at each one we will add either another agent $\psi_{best,j}$ or agent ψ_i , where i is the current iteration. Let v_{i-1} be the current vote result. The result of v_{i-1} is either: (i) victory for a_{best} , (ii) tie between a_{best} and other options, (iii) defeat for a_{best} .

(i) If v_{i-1} is a victory for a_{best} , the new agent can change the result only when it votes in another option. Suppose a_l is an option that upon receiving one more vote will change a victory for a_{best} into a tie between a_{best} and a_l . Agent $\psi_{best,j}$ will vote in option a_l with probability p_l , while agent ψ_i will vote in option a_l with probability $p_l + \gamma_{i,l}$. Therefore, if v_{i-1} is such that one vote can change a victory for a_{best} into a tie between a_{best} and other options, agent ψ_i will have a higher probability of changing a victory for a_{best} into a tie between a_{best} and other options.

(ii) If v_{i-1} is a tie between a_{best} and other options, agent $\psi_{best,j}$ will break the tie in favor of a_{best} with probability p_1 while agent ψ_i with probability $p_1 - \epsilon_i$. Therefore, agent $\psi_{best,j}$ will have a higher probability of breaking the tie in favor of a_{best} . Moreover, if a_l is an option that is currently tied with a_{best} , agent $\psi_{best,j}$ will vote for a_l with probability p_l , while agent ψ_i with probability $p_l + \gamma_{i,l}$. Therefore, agent ψ_i will have a higher probability of changing a tie between a_{best} and other options into a defeat for a_{best} .

(iii) If v_{i-1} is a defeat for a_{best} , agent $\psi_{best,j}$ will vote for a_{best} with probability p_1 while agent ψ_i will vote for a_{best} with probability $p_1 - \epsilon_i$. Therefore, if v_{i-1} is such that one vote can change a defeat for a_{best} into a tie between a_{best} and other options, agent $\psi_{best,j}$ will have a higher probability of changing a defeat for a_{best} into a tie between a_{best} and other options.

In all three cases, agent $\psi_{best,j}$ leads to a higher increase in the probability of picking a_{best} than agent ψ_i . Therefore, up to any iteration i , copies of $\psi_{best,j}$ will have a higher probability of playing the best action than a diverse team. Hence, if $\psi_{best,j} = \psi_{best} \forall j$, then copies of the best agent ψ_{best} will be the best team in all world states, and therefore it will be the optimal team. Therefore, for a diverse team to perform better, at least one agent must have either a higher probability of taking the best action or a lower probability of taking a suboptimal action than ψ_{best} in at least one world state. ■

This theorem, however, only gives the necessary conditions for a diverse team to be stronger than a non-diverse team. The sufficient conditions will depend on which specific game the agents are playing. Basically, given the pdf of the agents for a set of world states, we can calculate the pdf of both the diverse team, and the team made of copies of the best agent. If the diverse team has a higher probability of taking the best action in a subset of the world states that is enough for it to play better, considering that it will

have a lower probability of taking the best action in the complementary subset, then the diverse team will play better than copies of the best agent.

3.2.1.2 Optimal Voting Rules

In my next theorem, I show that given some conditions, the optimal voting rule for a diverse team is to consider plurality voting, but break ties in favor of the strongest agent that participates in the tie. Basically, we have to assume that all agents are strong enough to contribute to the team, so no agent should be ignored. If there are harmful agents in the team, we can try to remove them until the conditions of the theorem are satisfied. Again, we consider a utility function with a value of 1 for the optimal action and 0 for the other actions. Given a team Φ with size n , our conditions are:

Assumption 1 *Weak agents do not harm*

For any subset of Φ with an even number of agents n' , and for a fixed world state ω_j , let $\phi'_{best,j}$ be the best agent of the subset. We divide the agents in 2 sets: *Weak* containing the $n'/2 - 1$ agents that have the lowest probability of taking the best action and the highest probability of taking a suboptimal action, and *Strong* containing the $n'/2$ agents that have the highest probability of playing the best action and the lowest probability of taking a suboptimal action (except for the best agent $\phi'_{best,j}$, that is in neither one of the sets). We assume that when all agents in *Weak* and $\phi'_{best,j}$ vote together in an option a_x , and all agents in *Strong* vote together in another option a_y , the probability of a_x being the best action is higher than the probability of a_y being the best action.

Assumption 2 *Strong agents are not overly strong*

Given a fixed world state ω_j , I assume that if m_1 agents voted in an action a_x and m_2 agents voted in an action a_y , the probability of a_x being the best action is higher than a_y being the best action, if $m_1 > m_2$. If there is a situation where the opinion of a set of agents always dominates the opinion of another set, we can try to remove the dominated agents until the assumption holds true.

Theorem 3.2.2 *The optimal voting rule for a team is to consider the vote of all agents, but break ties in favor of the strongest agent if the above assumptions are satisfied.*

Proof: By *Assumption 2* we know that we are looking for a tie-breaking rule, as the action chosen by most of the votes should always be taken. Let's consider the sets and the voting result described in the *Assumption 1*. Let $\langle p_1, \dots, p_k \rangle$ be the pdf of agent $\phi'_{best,j}$, and the pdf of the other agents of the subset be $\langle p_1 - \epsilon_i, p_2 + \gamma_{i,2}, \dots, p_k + \gamma_{i,k} \rangle$, $\gamma_{i,l} \geq 0 \forall l \in (2, k)$ and $\sum_{l=2}^k \gamma_{i,l} = \epsilon_i$. Let b be a rank in $(2, k)$. The probability of a_x being the best action is given by:

$P_1 = \zeta * (p_1) \prod_{i \in Weak} (p_1 - \epsilon_i) \prod_{t \in Strong} (p_b + \gamma_{t,b})$,
where ζ is a constant (according to the Bayes theorem).

While the probability that a_y is the best action is given by:

$$P_2 = \zeta * (p_b) \prod_{i \in Weak} (p_b + \gamma_{i,b}) \prod_{t \in Strong} (p_1 - \epsilon_t)$$

By *Assumption 1*, we have that $P_1 > P_2$. We can generate another voting pattern by making one agent ϕ_{weak} in *Weak* vote for a_y and one agent ϕ_{strong} in *Strong* vote for a_x . The probability of a_x being the best action will change to:

$$P'_1 = P_1 * \frac{(p_1 - \epsilon_{strong})(p_b + \gamma_{weak,b})}{(p_1 - \epsilon_{weak})(p_b + \gamma_{strong,b})}$$

While the probability of a_y being the best action will change to:

$$P'_2 = P_2 * \frac{(p_1 - \epsilon_{weak})(p_b + \gamma_{strong,b})}{(p_1 - \epsilon_{strong})(p_b + \gamma_{weak,b})}$$

As $(p_1 - \epsilon_{strong}) > (p_1 - \epsilon_{weak})$ and $(p_b + \gamma_{weak,b}) > (p_b + \gamma_{strong,b})$ by the *Assumption 1*, we have that $P'_1 > P_1$. Similarly, as $(p_1 - \epsilon_{weak}) < (p_1 - \epsilon_{strong})$ and $(p_b + \gamma_{strong,b}) < (p_b + \gamma_{weak,b})$ by the *Assumption 1*, we have that $P'_2 < P_2$.

Therefore, assuming that $P_1 > P_2$, we have that $P'_1 > P'_2$. Hence, for all modifications that can be generated by switching one of the agents, it is better to break ties in favor of the strongest agent. We can use all these voting patterns as a base and apply the same process recursively, to generate all possible voting patterns with a tie. Therefore, it will always be better to break ties in favor of the strongest agent.

Now I consider voting patterns with a tie between more than two options. Let's suppose that in this case breaking ties in favor of the strongest agent ($\phi'_{best,j}$) is not the optimal voting rule. Therefore, we should break the tie in favor of some option a_y . This implies that a_y has a higher probability of being the best action than a_x , the option chosen by the best agent. Now let's remove the agents that voted in all other options except a_x and a_y . This affects the probability of a_x and a_y being the best action in the same way. Therefore, we should still break ties in favor of option a_y . However, we already showed that when there are two options we should break ties in favor of the strongest agent. Hence, we should break the tie in favor of option a_x . So, by contradiction, we see that if there is a tie between more than two options we should still break ties in favor of the strongest agent.

If the strongest agent of the team is not one of the agents involved in the tie, we can ignore the opinion of the strongest agent according to *Assumption 2*, and break the tie in favor of the strongest agent from the ones involved in the tie, because *Assumption 1* applies to any subset of the agents. ■

An application may not have the knowledge of the pdf of the agents in individual world states. Therefore, I also study an approximation of the optimal voting rule, that break ties in favor of the strongest agent overall, instead of breaking ties in favor of the strongest agent in a given world state. In the next section we will see that both the optimal voting rule and our approximation improves the performance of a diverse team.

3.3 Results

3.3.1 Synthetic

I perform synthetic experiments using the quantal response (QR) model for the agents [McKelvey and Palfrey, 1995]. The quantal response model is a pdf from behavioral game theory to approximate how human beings (or non-rational players) behave while playing a game. It states that the probability of playing the best action is the highest, and it decays exponentially as the utility of the action gets worse. I use the QR model in my experiment, because it is a convenient way to represent non-rational agents with different strengths playing a game with a great number of options.

The pdf depends on a parameter, λ , that defines how rational (i.e., strong) is the agent. As λ gets higher, the agent provides a closer approximation to a perfect player. I define a λ_{ij} for each agent i and world state j .

I generated 1200 random teams of 4 agents, playing in 10 world states, and with 82 possible actions. I define each λ_{ij} as a random number in the interval $(0, 7)$, according to a uniform distribution. For each team, we can calculate the diversity and the average strength of the agents, according to the equations defined earlier. In Figure 3.1, we can see the performance of each team, as a function of diversity and the strength of its members. The strength of a team can be calculated after we generate the pdf of the team, by calculating the probability of all possible situations where the system would pick a particular ranking position. I assume that all world states are equally likely, hence the strength of a team is the average over all world states. I used a utility vector that gives a value close to 1 to the best action, and a low value to the other actions.

I performed a multiple linear regression for each voting rule. The following models were found: *simple*: $z = -0.09 + 1.48s + 0.45d$; *static*: $z = -0.03 + 1.36s + 0.55d$; *optimal*: $z = 0.09 + 0.92s + 1.29d$. The variable s is the average strength of the team members, d is the diversity of the team, and z is the strength of the team. The coefficient of multiple determination (R^2) of the models are 0.96, 0.81, 0.88, respectively.

As can be seen, both diversity and strength had a positive weight. This shows that groups with more diversity are stronger, given a fixed strength for their members. It is interesting to note that the impact of diversity increases as we change the voting

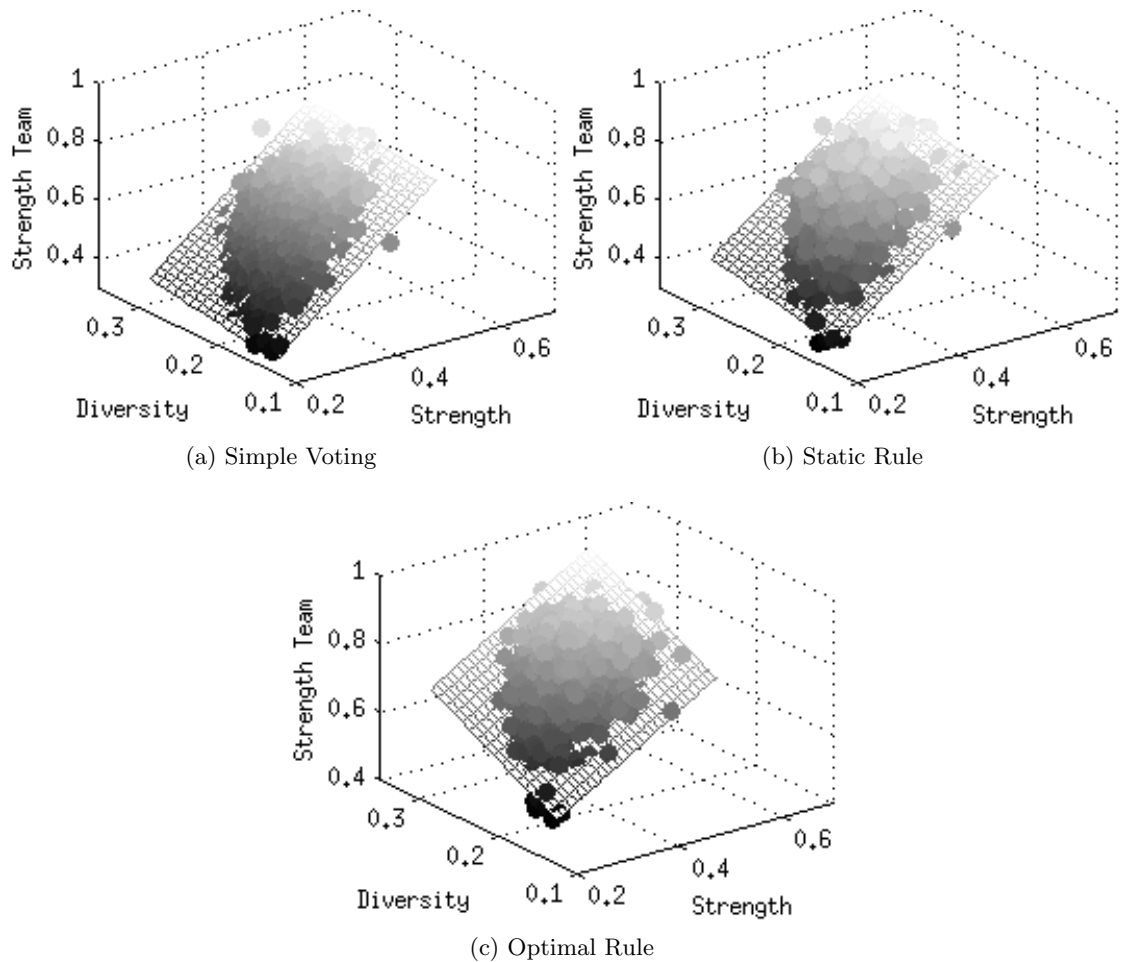


Figure 3.1: 1200 random teams of 4 agents.

rule from *simple* to *static*, and from *static* to *optimal*. The mean strength of all teams are $0.56(\pm 0.08)$, $0.61(\pm 0.08)$, $0.74(\pm 0.06)$, respectively. We can note that, as expected, *simple* had the lowest strength, followed by *static*, and *optimal* had the highest strength. The *optimal* voting rule is 30% stronger than *simple* voting in average.

3.3.2 Experiments in Computer Go

I also perform experiments with four Go software: Fuego 1.1, GnuGo 3.8, Pachi 9.01, MoGo 3, and two (weaker) variants of Fuego (Fuego Δ and Fuego Θ), in a total of 6 different agents. These are all publicly available Go software. Fuego is known to be the strongest Go software among all of them. Fuego, Pachi and MoGo follow a UCT Monte Carlo Go algorithm [Gelly et al., 2006]. Fuego uses heuristics to simulate games during the Monte Carlo simulations. There are mainly 5 possible heuristics in Fuego’s

code. These heuristics have a hierarchical order, and the original Fuego agent follows the order <Atari Capture, Atari Defend, Lowlib, Pattern> (The heuristic called Nakade is not enabled by default). Fuego Δ follows the order <Atari Defend, Atari Capture, Pattern, Nakade, Lowlib>. Fuego Θ follows the order <Atari Defend, Nakade, Pattern, Atari Capture, Lowlib>. The memory available for Fuego Δ and Fuego Θ is half of the memory available for Fuego.

All results presented are obtained by playing 1000 9x9 Go games, in a HP dl165 with dual dodeca core, 2.33GHz processors and 48GB of RAM. I first present results when my system plays as white, against the original Fuego playing as black with opening database. Then, I present results of my system playing as black, against the original Fuego playing as white with opening database. I will compare the winning rate of different agents and teams when playing against the same opponent. When I say that a result is significantly better than another, I use a *t-test* with 1% significance level ($\alpha = 0.01$).

I call a team composed by different Go software as “Diverse” or by the name of the voting rule that they use (“Simple” or “Static”). The team of copies of the strongest agent (Fuego) will be called “Uniform”. The copies are initialized with different random seeds, therefore due to the nature of the search algorithms, they will not always choose the same movement. When I want to be explicit about the number of agents in a team I will add a number after the name of the team. “Diverse” is composed by Fuego, GnuGo, Pachi and MoGo when executed with 4 agents, and is composed by all agents when executed with 6 agents. I also work with a parallelized version of Fuego (“Parallel”), and I will add a number after its name to indicate the number of threads.

Before introducing my results, I first analyze the agents under the classical voting theory and under my proposed theory. To simplify the analysis, I consider here the probability of playing the best move (P_{best}); therefore, I consider a utility vector with a value of 1 for the best move, and 0 for the other moves. I start by the classical voting theories. In order to estimate P_{best} , I use 1000 board states from my experiments. In 1000 games, I randomly choose a board state between the first and the last movement. I then ask Fuego to perform a movement in that state, but I give Fuego a time limit 50x higher than the default one. Therefore, Fuego is approximating how a perfect (or at least much stronger) player would play. To avoid confusion with the names I will call this agent Perfect. I then obtain Perfect’s evaluation for all the positions of the board, and organize them into a ranking.

I ran all agents in the selected 1000 board states and for each state I verify in which position of the ranking each agent would play. If, instead of playing, the agent resigns, I randomly pick a different board state and regenerate the data for all agents, including

Player	P_{best}	Team	P_{best}
Fuego	52.3%	Simple 4	57.5%
GnuGo	26.4%	Static 4	61.8%
Pachi	40.6%	Uniform 4	79.6%
MoGo	40.8%	Simple 6	71.1%
Fuego Δ	48.8%	Static 6	72.4%
Fuego Θ	47.7%	Uniform 6	86.6%

(a) Players (b) Teams

Table 3.3: Probability to select the best move of each player and the teams.

Perfect’s evaluation. Based on that, I can generate a histogram for all agents, shown in Figure 3.2.

Assuming that the agents are independent, and that each one will choose a move according to the probability distribution corresponding to its histogram, we can calculate P_{best} of any group and voting rule that we want. Basically we have to calculate the probability of all the possible situations where the system would pick the best move. For a team of k agents we have to calculate $O(n^{k-1})$ probabilities, where n is the number of possible options. While for a team of 4 agents I am able to calculate the precise value, for a team of 6 agents I am going to show approximations.

In Table 3.3 we can see P_{best} of each individual player and of all teams. The P_{best} of the teams is higher than the P_{best} of each one of the agents, and is higher for a team of 6 agents than for a team of 4 agents. This result is expected when we consider the extended version of the Condorcet’s Jury Theorem [List and Goodin, 2001], at least for a uniform team. According to the theorem P_{best} approaches 1 when the number of agents goes to infinity. However, we would also expect Uniform to perform better than Diverse. Would it be possible, then, for a diverse team to perform better than a uniform team?

Intuitively, we would expect that a uniform team would agree on certain moves much more often than a diverse team. And indeed, when we look at the graph of the frequency of the size of the set of agents that voted for the winning move (Figure 3.3), we can see that they are very different. In the x-axis I show the number of agents that agreed on the selected movement, and in the y-axis the frequency of each number considering all moves in the 1000 games. The expected size of the set for Diverse is 3.50, while for Uniform is 4.43. Therefore, if Fuego plays badly in a certain board state, all copies of Fuego would also tend to vote for the same bad moves. In a diverse team, however, some agents could be able to play better in that particular situation. The extended Condorcet’s Jury Theorem assumes that agents are independent, but in fact their relative performances might change according to the state of the board.

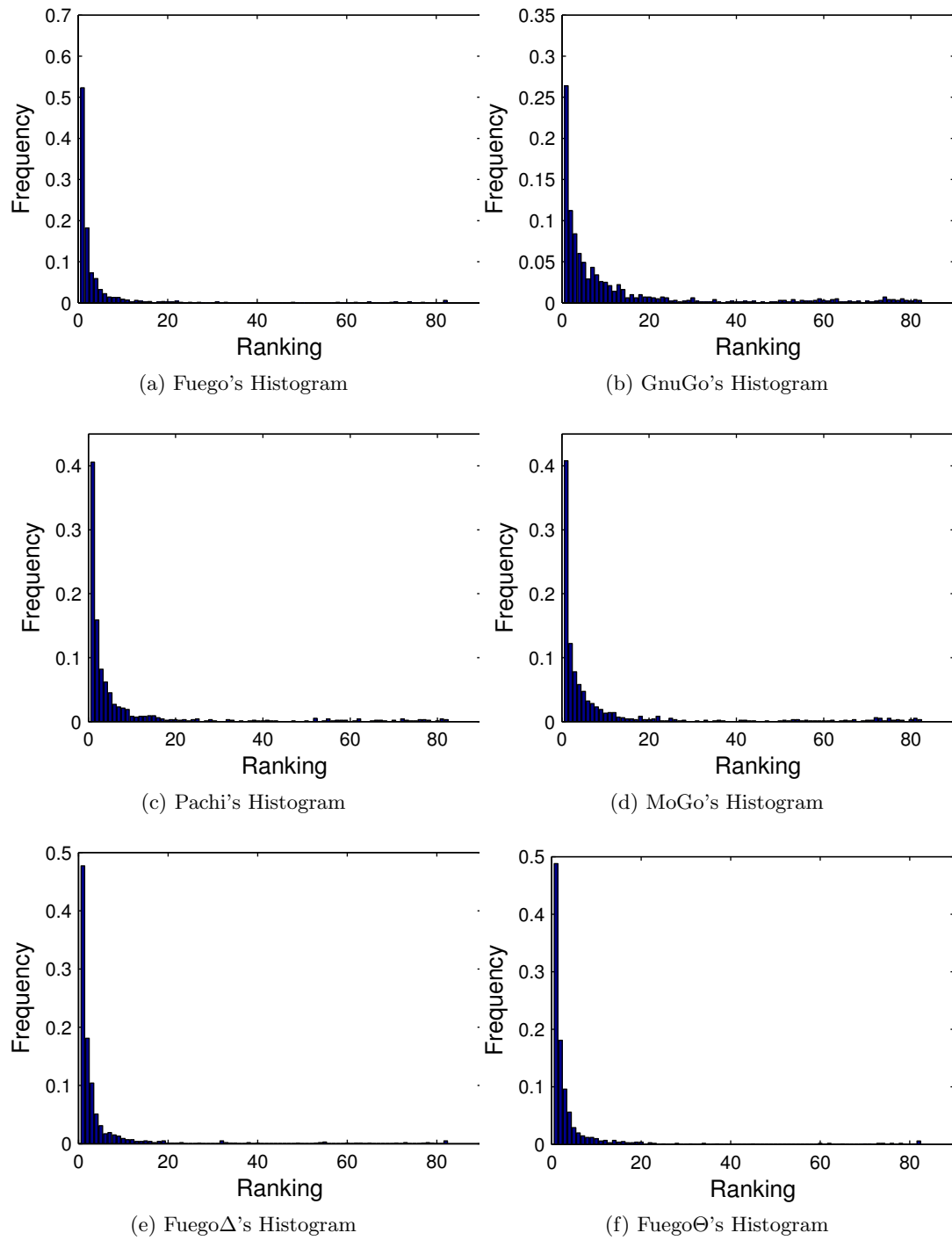


Figure 3.2: Histogram of the agents, using real data.

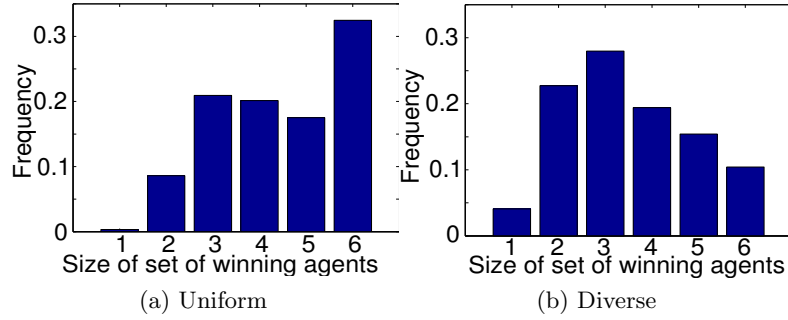


Figure 3.3: Expected size of the set of agents that vote for the winning move, with 6 agents and no opening database.

Player	# Higher P_{best}
GnuGo	17% (12%)
Pachi	21% (11%)
MoGo	20% (7%)
Fuego Δ	25% (6%)
Fuego Θ	26% (6%)

Table 3.4: Weak agents can play better in some board states. In parentheses, I show when the difference in P_{best} is 99% significant.

Now I analyze the agents according to my proposed theory. I will use Theorem 3.2.1 to justify that it is worth it to explore a diverse team. If Fuego, the strongest agent, is always stronger in all board positions, then we can just use copies of Fuego as the optimal team. Therefore, I will test if all agents are able to play better than Fuego in some board positions. I selected 100 board states, and I played all agents 50 times for each board state. Based on my estimate of the best move (obtained from Perfect), I can calculate P_{best} for each agent and for each board state. In Table 3.4, we can see in how many board states the agents have a higher P_{best} than Fuego (in its default time limit). As can be observed, all agents are able to play better than Fuego in some board positions, therefore it is possible for a diverse team to play better than copies of the best agent. As the number of board states where an agent plays better is not small, we can expect that a diverse team should be able to overcome the uniform team.

According to Theorem 3.2.2, if we assume that the weak agents (like GnuGo) are not weak enough to harm the system, and the strong agents (like Fuego and its variants) are not strong enough to dominate a subset of the agents, then the optimal voting rule is to break ties in favor of the strongest agent. However, during a game the system does not have access to the pdf of the agents, and has no way to identify which is the strongest agent. Therefore, I present results using the *static* voting rule, that break ties in favor

of the strongest agent overall. Based on my synthetic results, we can predict that *static* should perform better than *simple*. I also tried a weighted voting rule, which allowed me to empirically learn the best weights by a hill climbing algorithm. The resulting rule was equivalent to the *static* voting rule.

We can see my results for white in Figure 3.4(a,b). Diverse plays significantly better than Fuego, with 6 agents or with the *static* voting rule. When I keep the opening database, Diverse plays significantly better than Uniform and Parallel with 6 agents. Without the opening database, Diverse still plays significantly better than Parallel with 6 agents, but the difference between Diverse and Uniform is not significant. Static is either significantly better than Simple, or the difference between them is not significant.

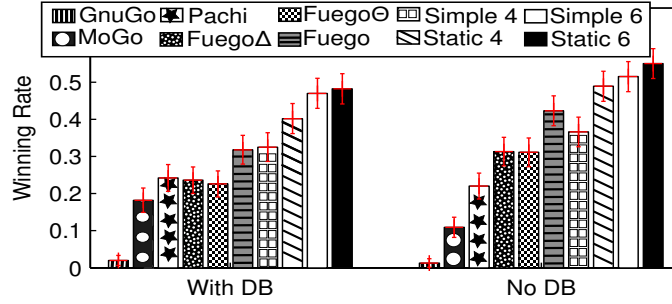
In Figure 3.4(c,d) we can see the results for black. Again, Diverse plays significantly better than Fuego when using the *static* voting rule. This time, however, Diverse (with 6 agents or using the *static* voting rule) is able to play significantly better than Uniform without the opening database, but with the opening database the difference between them is not significant. Again, Static is either significantly better than Simple, or the difference between them is not significant. Static is always significantly better than Parallel.

To verify the generality of improving the results by the *static* voting rule and by adding more agents, I also played our system as white against Pachi as black, without opening database. Simple 4 won 56.2% of the games, Static 4 won 65.5% and Simple 6 won 66.8%. Therefore, these techniques can improve the results in other situations.

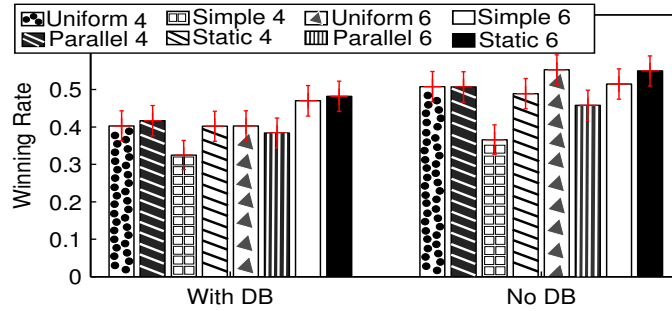
By the classical view of voting, my experimental result is not expected. If we view each agent as having a fixed pdf, we would predict that copies of the best agent would perform much better than a diverse team with weaker agents. However, in my results I showed that the diverse team has a competitive strength, and is able to play even better than copies of the best agent in some situations. My new model provides a theoretical explanation for my experimental results.

3.4 Detailed Study: Why a Team of Diverse Agents Perform Better?

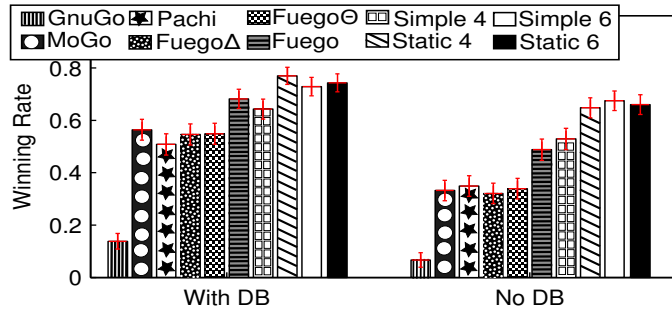
I study in detail three games from my experiments, in order to better understand why a team of weak players can perform as well, or better than a team made of copies of the best player. I study games with 6 agents, using the *simple* voting rule. According to my theoretical work, at least one agent must play better than the strongest agent in at least one world state for a diverse team to overcome a uniform team. These are only necessary conditions, for a diverse team to effectively play better this must happen in many world states, specially in critical situations that can decide the game. Here I show that this



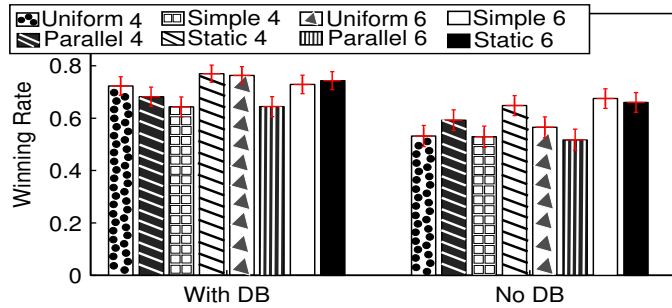
(a) Results for white. Single agents and the diverse team (Simple/Static).



(b) Results for white. The uniform team, the diverse team (Simple/Static), and a parallelized agent (Parallel).



(c) Results for black. Single agents and the diverse team (Simple/Static).



(d) Results for black. The uniform team, the diverse team (Simple/Static), and a parallelized agent (Parallel).

Figure 3.4: Results in the Computer Go domain. The error bars show the confidence interval, with 99% of significance.

really happens in Computer Go, based on an analysis by an expert human player. As Go is a complex game, note that some expert readers might not agree completely with all points of this analysis. Although I present results in the Computer Go domain, this phenomenon should also occur in other complex domains, where the relative strength of the agents change according to the world state.

These games are analyzed by Chao Zhang, a 4-dan amateur Go player. In order to show that the weak agents are not playing better simply by chance, I estimate the probability of all agents playing all analyzed moves by repeatedly playing them 100 times in the board state under consideration. Based on these probabilities, I calculate the probabilities of the diverse team and the uniform team, to show that the diverse team would perform better in these board states. An important point to note is that it is not the case that a certain subset of the agents always vote for a better move; the set of agents that can find a better move than Fuego changes according to each board state.

This analysis requires some Go knowledge to be fully understood. Go is a turn-based game between two players: black and white. At each turn, the players must place a stone in an empty intersection of the board. If a group of stones is surrounded by the opponent's stones they are removed from the board (i.e. they are "killed"). The stones that surround an area form a territory, whose value is counted by the number of empty intersections inside. In the end of the game, the score is defined by the amount of territory minus the number of captured stones, and the player with the highest score wins. A detailed description of the rules can be found in Pandanet [2016].

I first analyze the Go game in Figure 3.5. In some positions, the weak agents vote for better moves than Fuego, the strongest agent. Move 11 is a very interesting situation. Here, Fuego, Pachi and MoGo vote for move D4, while GnuGo votes for E8 (X). Even though GnuGo is the weakest agent, in this situation it is able to find a better move than all other agents. E8 is better because it allows white to get the territory in the upper left corner. Besides, white can aim at G7 to kill the black group in the upper right. If white plays D4, black can play E8 to kill white aiming at the upper left corner. Unfortunately, GnuGo loses the vote in this situation. In all other positions, I show situations where the weak agents vote together for a better move than Fuego. For example, in move 23, Fuego votes for B4 (Δ) while Pachi, MoGo, Fuego Δ and Fuego Θ vote for B7. If white chooses B7, white can kill C7&D7 or B5&C5&D5&E5. If black saves C7&D7, white can use B4 to kill the other group; If black saves B5&C5&D5&E5, white can use C8 to kill C7&D7. If white chooses B4, black will use B7 to kill the white group in the upper left. Fuego's mistake is critical in this situation, and would lead to losing the game. In move 45, Fuego would make another mistake. Fuego votes for B9 (Δ), while GnuGo and Pachi vote for H3. B9 wastes a move: it cannot affect the final result and wastes a chance for further

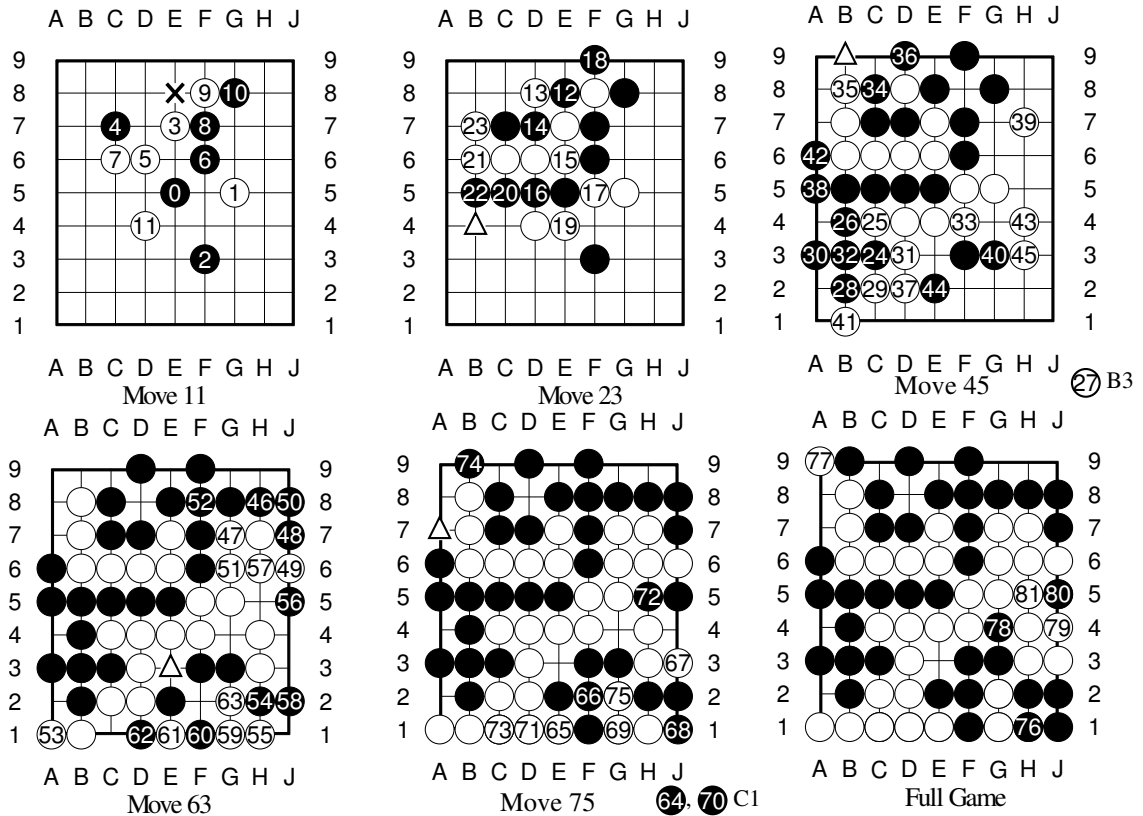


Figure 3.5: First example, the diverse team plays as white without the opening database against Fuego. White wins by resignation.

developments. H3, on the other hand, is aiming at killing black in the right bottom. In this case Fuego Δ also votes for B9, so the *static* voting rule would choose a worse move. In move 63, Fuego would play E3, while GnuGo, Pachi, Fuego Δ and Fuego Θ vote for G2. If white plays G2, the black group in the right bottom dies, while if white plays E3, white cannot kill them. This is another critical mistake, that would make white lose the game. Finally, in move 75, Fuego votes for A7, while Pachi and Fuego Θ vote for G2. G2 is better than A7, as it allows white to have a larger territory. As can be seen, there are many situations where the weaker agents vote together for a better move than Fuego. The probabilities of each agent playing the analyzed moves can be seen in Table 3.5. It is clear that Fuego did not choose the worse move by accident: in many cases it has a lower probability than the other agents of playing the best move between the two options. Consequently, the uniform team is still not able to perform well in these situations, it still has a low probability of playing the best move, and it is always outperformed by the diverse team. In some situations, the probability of playing the worst move even increases by using multiple copies of Fuego.

Agent	Move 11		Move 23		Move 45		Move 63		Move 75	
	E8*	D4	B7*	B4	H3*	B9	G2*	E3	G2*	A7
Fuego	2%	51%	83%	14%	1%	76%	53%	29%	22%	16%
GnuGo	100%	0%	0%	0%	100%	0%	100%	0%	0%	0%
Pachi	6%	75%	30%	70%	46%	0%	78%	1%	35%	1%
MoGo	2%	61%	100%	0%	0%	0%	0%	84%	53%	0%
Fuego Δ	24%	19%	100%	0%	16%	19%	76%	13%	24%	7%
Fuego Θ	35%	9%	99%	0%	12%	30%	78%	10%	31%	11%
Diverse	15%	57%	99%	0%	20%	28%	88%	7%	45%	5%
Uniform	0%	73%	95%	4%	0%	98%	63%	26%	23%	21%

Table 3.5: Probability of playing the moves in the first example. * indicates the better move.

I now analyze the game in Figure 3.6. In move number 4, Fuego Δ and Pachi vote for C7, while Fuego votes for move G3 (Δ). G3 is a bad opening for white, because the two white groups would be split by black. Another example is in move 7, when GnuGo, Pachi and Fuego Δ vote for B6, while Fuego votes for G7 (Δ). Black and white are fighting in the upper left corner. If white plays G7, it waives the fight and plays in a place that is not immediately important. White should choose B6 to continue the fight in order to win. Even GnuGo, the weakest agent, knows that B6 is a better move. In move 25, GnuGo and Mogo choose A8, while Fuego chooses F2 (Δ). If white does not play A8, black will play A5 to kill the white group in the left side. White has to kill with A8. This time Fuego’s mistake is critical, and could lead to losing the whole game. In this situation GnuGo helps avoid a critical mistake, because Fuego Θ also votes for F2. Moreover, it is an example of a case where the *static* voting rule fails, as it would break the tie in favor of Fuego. I expect that significant improvements in game play would be possible if we learn which is the strongest agent in a given situation, and better approximate the *optimal* voting rule. Another interesting move is 37. Fuego and Fuego Δ vote for D2, while MoGo and Fuego Θ vote for E3. Both moves are equally good, as they get the same territory. However, GnuGo might have a better move: F6. If white plays F6, it can aim at both G6 and F4 for the next moves, which will cause great harm to black’s territory. This is another example of a situation where the weakest agent has a better move than all other agents. The probabilities of each agent playing the analyzed moves can be seen in Table 3.6. Again, we can see that the diverse team would have a higher probability of finding the better moves than the uniform team.

In the games with the opening database, an interesting one is in Figure 3.7. In move 29, GnuGo, Pachi and MoGo choose D2, while Fuego votes for B8 (Δ). D2 can protect the lower left, while B8 cannot kill the black group in the upper left, and ends up making

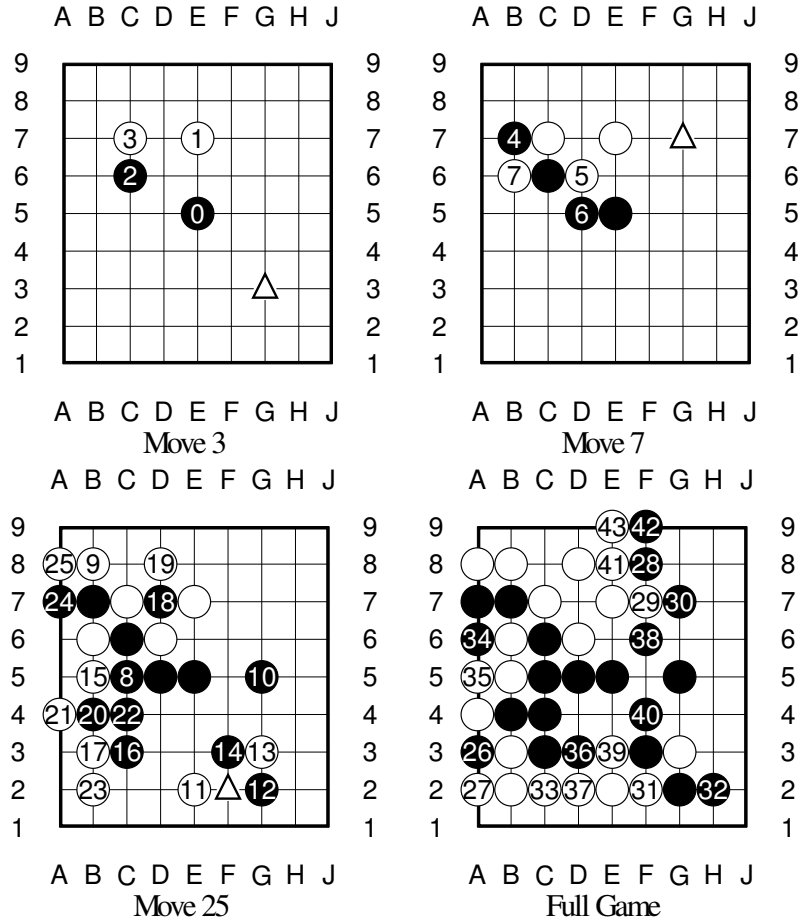


Figure 3.6: Second example, the diverse team plays as white without the opening database against Fuego. White wins by resignation.

Agent	Move 3		Move 7		Move 25		Move 37		
	C7*	G3	B6*	G7	A8*	F2	F6*	D2	E3
Fuego	20%	2%	7%	41%	11%	30%	1%	53%	19%
GnuGo	0%	0%	100%	0%	100%	0%	100%	0%	0%
Pachi	27%	14%	99%	1%	28%	19%	26%	27%	0%
MoGo	0%	8%	1%	0%	89%	0%	0%	41%	45%
Fuego Δ	20%	0%	34%	20%	28%	10%	0%	83%	7%
Fuego Θ	25%	4%	50%	7%	37%	11%	0%	80%	12%
Diverse	–%	–%	77%	0%	70%	1%	1%	90%	5%
Uniform	19%	0%	2%	30%	7%	27%	0%	84%	8%

Table 3.6: Probability of playing the moves in the second example. Some results are unavailable due to lack of memory. * indicates the better move.

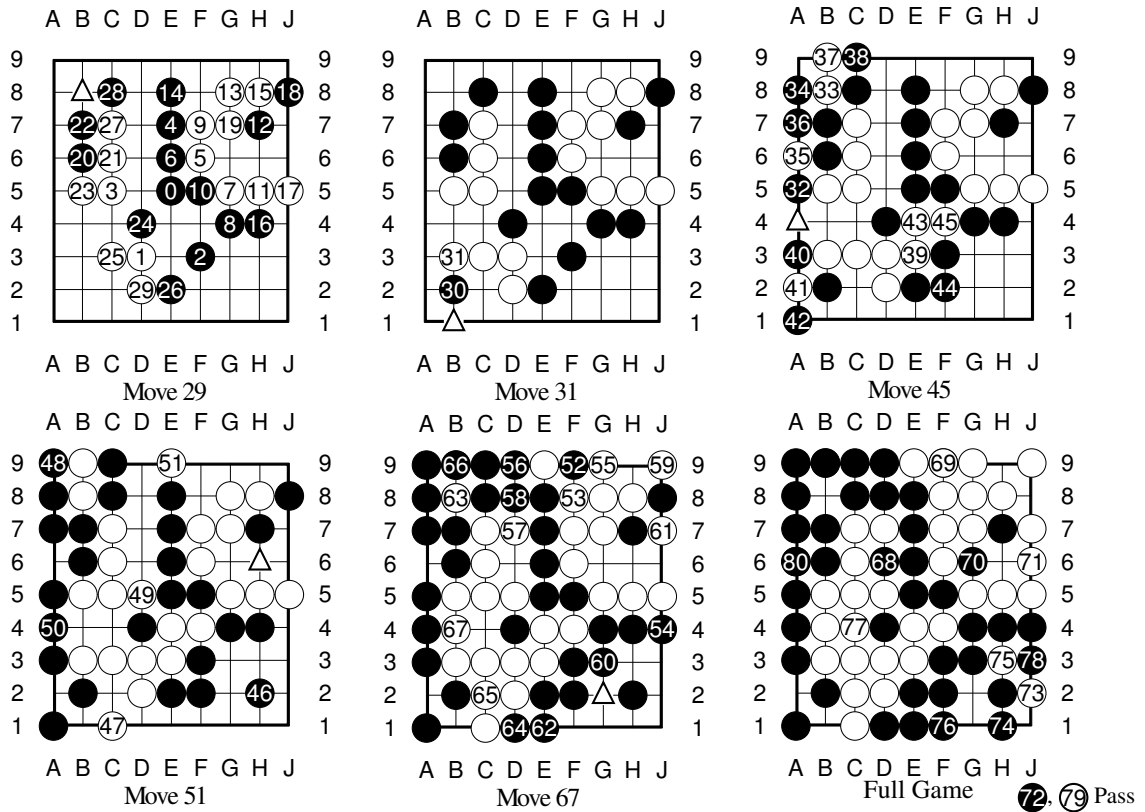


Figure 3.7: Third example, the diverse team plays as white with the opening database against Fuego. White wins by resignation.

it more solid. In move 31, Pachi and MoGo vote for B3, and Fuego votes for B1 (Δ). Even if both moves might be able to kill the black stone in B2, B3 can kill it for sure. If white plays B1, black can play B3 and would lead to complications. This mistake could make white lose the game. If black survives, it can kill the white group in the lower left. In move 45, Pachi, MoGo and Fuego Δ vote for F4, while Fuego votes for A4 (Δ). F4 splits black into two groups and can make use of this division in the future. A4 just wastes a move and gives black more territory. In move 51, Pachi and MoGo choose E9, and Fuego chooses H6 (Δ). E9 makes the white group on the left survive, while H6 wastes a move and will lead to the death of the white group. This is a critical mistake, that would make white lose the game. In move 67, GnuGo and MoGo vote for B4, while Fuego votes for G2 (Δ). B4 is better, as it can get more territory. G2 just wastes a move. The probabilities of each agent playing the analyzed moves can be seen in Table 3.7. Again, in all these situations the diverse team has a higher probability of playing the better move than the uniform team. In some cases, the probability of playing the worse move even increases with multiple copies of Fuego.

Agent	Move 29		Move 31		Move 45		Move 51		Move 67	
	D2*	B8	B3*	B1	F4*	A4	E9*	H6	B4*	G2
Fuego	3%	16%	44%	26%	17%	40%	0%	35%	0%	12%
GnuGo	100%	0%	0%	0%	0%	0%	0%	0%	100%	0%
Pachi	77%	18%	64%	0%	78%	17%	90%	0%	3%	0%
MoGo	91%	0%	98%	0%	92%	0%	51%	0%	46%	4%
Fuego Δ	6%	4%	11%	1%	51%	12%	0%	1%	0%	9%
Fuego Θ	5%	7%	13%	0%	50%	21%	0%	2%	0%	5%
Diverse	82%	3%	75%	0%	54%	12%	37%	0%	9%	1%
Uniform	0%	12%	56%	32%	5%	53%	0%	44%	0%	4%

Table 3.7: Probability of playing the moves in the third example. * indicates the better move.

3.5 Conclusion and Discussions

I showed that diverse teams can outperform teams composed by copies of the best player. However, it is still a challenge to find the best possible teams. In an open multi-agent system the pdfs of the agents are generally not available. Moreover, in many complex scenarios we cannot even easily enumerate all the possible states of the world. Hence, given a world state, how can we quickly and automatically know the relative strength of the different agents? This is still an important open problem.

I gave an initial step by studying in detail different scenarios where diverse agents are able to outperform the best agent. One possible direction for future work is to identify common characteristics of world states where a certain agent is able to play better than the best agent. Hence, given a new world state we would be able to estimate the strongest agent for that specific world state and better approximate the *optimal* voting rule. In addition, we could also dynamically change the team in order to have the best (or close to the best) possible one for each different scenario.

In real-life scenarios, like robot teams, the problem is even more challenging. We can always estimate the pdf of an agent by running it multiple times in a given world state, if we have at least an estimation of the ground truth. However, for an embodied agent, the number of times we can sample might be very limited. A similar challenge is faced in Evolutionary Robotics [Nolfi and Floreano, 2001], where a great range of robots/controllers must be constantly evaluated. One common approach is to perform the evaluation in simulation, and implement in real life the best performing solution. Likewise, we could sample the pdf of different robots in simulation, in order to estimate their pdfs in the real world. Of course, the accuracy of the pdf estimation would depend on the accuracy of the simulation environment.

In general, however, even without knowledge of the pdfs of the agents, this chapter shows that a team composed by strong but very similar agents is not necessarily optimal. Hence, if an operator is not able to estimate the pdfs, she should at least evaluate the performance of diverse teams before picking only the strongest agents as the chosen team for a certain multi-agent application.

In the next chapter I will present a second model of diversity, which will allow an operator to better identify in which situations diverse teams should be preferred.

Chapter 4

GIVE A HARD PROBLEM TO A DIVERSE TEAM

Together, we form a necessary paradox; not a senseless contradiction.
(Criss Jami)

Team formation is crucial when deploying a multi-agent system [Nair and Tambe, 2005, Guttman, 2008, Liemhetcharat and Veloso, 2012, Matthews et al., 2012]. Many researchers emphasize the importance of diversity when forming teams [LiCalzi and Surucu, 2012, Lamberson and Page, 2012, Hong and Page, 2004]. However, there are many important questions about diversity that were not asked before, and are not explored in such models.

LiCalzi and Surucu [2012] and Hong and Page [2004] propose models where the agents know the utility of the solutions, and the team converges to the best solution found by one of its members. Clearly in complex problems the utility of solutions would not be available, and agents would have to resort to other methods, such as voting, to take a common decision. Lamberson and Page [2012] study diversity in the context of forecasts, where the solutions are represented by real numbers and the team takes the average of the opinion of its members. Domains where the possible solutions are discrete, however, are not captured by such a model.

In the previous chapter, I studied teams of agents that vote in discrete solution spaces. I showed that a diverse team of weaker agents can overcome a uniform team made of copies of the best agent. However, this does not always occur, and the previous model do not present ways to know when we should use diverse teams. Moreover, it lacks a formal study of how the performance of diverse teams change as the number of agents and/or actions increases.

In this chapter I shed new light on this problem, by presenting a new, more general model of diversity for teams of voting agents. My model captures better than the previous ones the notion of a diverse team as a team of agents that tend to not agree on the same

actions, and allows us to make new predictions. My main insight is based on the notion of *spreading tail* (*ST*) and *non-spreading tail* (*NST*) agents. As I will show, a team of *ST* agents has a diverse behavior, i.e., they tend to not agree on the same actions. Hence, I can model a diverse team as a team of *ST* agents, and show that the performance improves as the size of the action space gets larger. I also prove upper and lower bounds on how fast different teams converge. The improvement can be large enough to overcome a uniform team of *NST* agents, even if individually the *ST* agents are weaker. As it is generally hard to find good solutions for problems with a large number of actions, it is important to know which teams to use in order to tackle such problems. Moreover, I show that the performance of a diverse team converges to the optimal one exponentially fast as the team grows. My synthetic experiments provide even further insights about my model: even though the diverse team overcomes the uniform team in a large action space, the uniform team eventually will again play better than the diverse team as the action space keeps increasing if the best agent does not behave exactly like an *NST* agent.

Finally, I test my predictions by studying a system of voting agents, in the Computer Go domain. I show that a uniform team made of copies of the best agent plays better in smaller board sizes, but is overcome by a diverse team as the board gets larger. Moreover, I analyze the agents and verify that weak agents have a behavior closer to *ST* agents, while the best agent is closer to an *NST* agent. Therefore, I show that my predictions are verified in a real system, and can effectively be used while forming a multi-agent team.

4.1 Model for Analysis of Diversity in Teams

Consider a problem defined by choosing an action a from a set of possible actions \mathbf{A} . Each a has an utility $U(a)$, and our goal is to maximize the utility. I always list the actions in order from best to worst, therefore $U(a_j) > U(a_{j+1}) \forall j$ (a_0 is the best action). In some tasks (like in Section 4.2), a series of actions are chosen across different states, but here I focus on the decision process in a given state.

Consider a set of agents, voting to decide over actions. The agents do not know the utility of the actions, and vote for the action they believe to be the best according to their own decision procedure, characterized by a probability distribution (pdf). I write as $p_{i,j}$ the probability of agent i voting for action a_j . I denote by $p_{i,j}(m)$, when I explicitly talk about $p_{i,j}$ for an action space of size m . If the pdf of one agent is identical to the pdf of another agent, they will be referred to as copies of the same agent. The action that wins by plurality voting is taken by the team. Ties are broken randomly, except when I explicitly talk about a tie breaking rule. Let \mathbf{D}_m be the set of suboptimal actions ($a_j, j \neq 0$) assigned with a nonzero probability in the pdf of an agent i , and $d_m = |\mathbf{D}_m|$. I assume that

there is a bound in the ratio of the suboptimal action with highest probability and the one with lowest nonzero probability, i.e., let $p_{i,min} = \min_{j \in \mathbf{D}_m} p_{i,j}$ and $p_{i,max} = \max_{j \in \mathbf{D}_m} p_{i,j}$; there is a constant α such that $p_{i,max} \leq \alpha p_{i,min} \forall$ agents i .

I define strength as the expected utility of an agent and/or a team. The probability of a team playing the best action will be called p_{best} . I first consider a setting where $U(a_0) \gg U(a_j) \forall j \neq 0$, hence I can use p_{best} as my measure of performance. I will later consider more general settings, where the first r actions have a high utility.

I define team formation as selecting from the space of all agents a limited number of agents that has the maximum strength by voting together to decide on actions. I study the effect of increasing the size m of the set of possible actions on the team formation problem. Intuitively, the change in team performance as m increases will be affected by how the pdf of the individual agents i change when m gets higher. As we increase m , d_m can increase or not change. Hence, I classify the agents as *spreading tail* (*ST*) agents or *non-spreading tail* agents (*NST*).

I define *ST* agents as agents whose d_m is non-decreasing on m and $d_m \rightarrow \infty$ as $m \rightarrow \infty$. I consider that there is a constant $\epsilon > 0$, such that for all *ST* agents i , $\forall m$, $p_{i,0} \geq \epsilon$. I assume that $p_{i,0}$ does not change with m , although later I discuss what happens when $p_{i,0}$ changes.

I define *NST* agents as agents whose pdf does not change as the number of actions m increases. Hence, let m_{i0} be the minimum number of actions necessary to define the pdf of an *NST* agent i . We have that $\forall m, m' \geq m_{i0}, \forall j \leq m_{i0} p_{i,j}(m) = p_{i,j}(m'), \forall j > m_{i0} p_{i,j}(m) = 0$.

I first give an intuitive description of the concept of diversity, then define formally diverse teams. By diversity, I mean agents that tend to disagree. In the previous chapter, a diverse team is defined as a set of agents with different pdfs. Hence, they disagree because of having different probabilities of playing certain actions. In this chapter, I generalize the previous definition to capture cases where agents disagree on actions, regardless of whether their pdfs are the same or not. Formally, I define a diverse team to be one consisting of a set of *ST* agents (either different *ST* agents or copies of the same *ST* agent). In my theoretical development I will show that this definition captures the notion of diversity: a team of *ST* agents will tend to not agree on the same suboptimal actions. I call uniform team as the team composed by copies of an *NST* agent. This is an idealization to perform my initial analysis. I will later discuss more complex domains, where the agents of the uniform team also behave like *ST* agents.

Agents	Action 1	Action 2
Agent 1	0.6	0.4
Agent 2	0.55	0.45
Agent 3	0.55	0.45
Uniform p_{best} :	0.648	
Diverse p_{best} :	0.599	

(a) With 2 actions, uniform team plays better than diverse team.

Agents	Action 1	Action 2	Action 3
Agent 1	0.6	0.4	0
Agent 2	0.55	0.25	0.2
Agent 3	0.55	0.15	0.3
Uniform p_{best} :	0.648		
Diverse p_{best} :	0.657		

(b) When we add one more action, diverse team plays better than uniform team.

Table 4.1: Performance of diverse team increases when the number of actions increases.

4.1.1 A Hard Problem to a Diverse Team

I start with an example, to give an intuition about my model. Consider the agents in Table 4.1(a), where I show the pdf of the agents, and p_{best} of the uniform team (three copies of agent 1) and the diverse team (one copy of each agent). I assume agent 1 is an *NST* agent, while agent 2 and 3 are *ST* agents. In this situation the uniform team plays better than the diverse team. Now let's add one more action to the problem. Because agent 2 and 3 are *ST* agents, the probability mass on action 2 scatters to the newly added action (Table 4.1(b)). Hence, while before the *ST* agents would always agree on the same suboptimal action if they both did not vote for the optimal action, now they might vote for different suboptimal actions, creating a tie between each suboptimal action and the optimal one. Because ties are broken randomly, when this happens there will be a 1/3 chance that the tie will be broken in favor of the optimal action. Hence, p_{best} increases when the probability of the *ST* agents agreeing on the same suboptimal actions decreases, and the diverse team now plays better than the uniform team, even though individually agents 2 and 3 are weaker than agent 1.

I now present my theoretical work. First I show that the performance of a diverse team converges when $m \rightarrow \infty$, to a value that is higher than the performance for any other m .

Theorem 4.1.1 $p_{best}(m)$ of a diverse team of n agents converges to a certain value \tilde{p}_{best} as $m \rightarrow \infty$. Furthermore, $\tilde{p}_{best} \geq p_{best}(m)$, $\forall m$.

Proof: Let $p_{i,min} = \min_{j \in \mathbf{D}_m} p_{i,j}$, $p_{i,max} = \max_{j \in \mathbf{D}_m} p_{i,j}$ and \mathbf{T} be the set of agents in the team. By my assumptions, there is a constant α such that $p_{i,max} \leq \alpha p_{i,min}$ for all agents i . Then, we have that $1 \geq 1 - p_{i,0} = \sum_{j \in \mathbf{D}_m} p_{i,j} \geq d_m p_{i,min}$. Therefore, $p_{i,min} \leq \frac{1}{d_m} \rightarrow 0$ as d_m tends to ∞ with m . Similarly, $\alpha p_{i,min} \rightarrow 0$ as $d_m \rightarrow \infty$. As $p_{i,j} \leq \alpha p_{i,min}$ we have that $\forall j p_{i,j} \rightarrow 0$ as $d_m \rightarrow \infty$. I show that this implies that when $m \rightarrow \infty$, weak agents never agree on the same suboptimal action. Let i_1 and i_2 be two arbitrary agents. Without loss of generality, assume i_2 's d_m ($d_m^{(i_2)}$) is greater than or equal i_1 's d_m ($d_m^{(i_1)}$). The probability (σ_{i_1,i_2}) of i_1 and i_2 agreeing on the same suboptimal action is upper bounded by $\sigma_{i_1,i_2} = \sum_{a_j \in \mathbf{A} \setminus a_0} p_{i_1,j} p_{i_2,j} \leq d_m^{(i_2)} p_{i_1,max} p_{i_2,max} \leq d_m^{(i_2)} \alpha p_{i_2,min} p_{i_1,max} \leq \alpha p_{i_1,max}$ (as $d_m^{(i_2)} p_{i_2,min} \leq 1$). We have that $\alpha p_{i_1,max} \rightarrow 0$ as $p_{i_1,max} \rightarrow 0$, because α is a constant. Hence the probability of any two agents agreeing on a suboptimal action is $\frac{\sum_{i_1 \in \mathbf{T}} \sum_{i_2 \in \mathbf{T}, i_2 \neq i_1} \sigma_{i_1,i_2}}{2} \leq \frac{n(n-1)}{2} \max_{i_1,i_2} \sigma_{i_1,i_2} \rightarrow 0$, as n is a constant.

Hence, when $m \rightarrow \infty$, the diverse team only chooses a suboptimal action if all agents vote for a different suboptimal action or in a tie between the optimal action and suboptimal actions (because ties are broken randomly). Therefore, p_{best} converges to:

$$\tilde{p}_{best} = 1 - \prod_{i=1}^n (1 - p_{i,0}) - \sum_{i=1}^n (p_{i,0} \prod_{j=1, j \neq i}^n (1 - p_{j,0})) \frac{n-1}{n}, \quad (4.1)$$

that is, the total probability minus the cases where the best action is not chosen: the second term covers the case where all agents vote for a suboptimal action and the third term covers the case where one agent votes for the optimal action and all other agents vote for suboptimal actions.

When m is finite, the agents might choose a suboptimal action by agreeing over that suboptimal action. Therefore, we have that $p_{best}(m) \leq \tilde{p}_{best} \forall m$. ■

Let $p_{best}^{uniform}(m)$ be p_{best} of the uniform team, with m actions. A uniform team is not affected by increasing m , as the pdf of an *NST* agent will not change. Hence, $p_{best}^{uniform}(m)$ is the same, $\forall m$. If \tilde{p}_{best} is high enough so that $\tilde{p}_{best} \geq p_{best}^{uniform}(m)$, the diverse team will overcome the uniform team, when $m \rightarrow \infty$. Therefore, the diverse team will be better than the uniform team when m is large enough.

In practice, a uniform team made of copies of the best agent might not behave exactly like a team of *NST* agents, as the best agent could also increase its d_m as m gets larger. I discuss this situation in Section 4.2. In order to perform that study, I derive in the following corollary how fast p_{best} converges to \tilde{p}_{best} , as a function of d_m .

Corollary 4.1.2 $p_{best}(m)$ of a diverse team increases to \tilde{p}_{best} in the order of $O(\frac{1}{d_m^{min}})$ and $\Omega(\frac{1}{d_m^{max}})$, where d_m^{max} is the highest and d_m^{min} the lowest d_m of the team.

Proof: I assume here the notation that was used in the previous proof. First I show a lowerbound on $p_{best}(m)$. We have that $p_{best}(m) = 1 - \psi_1$, where ψ_1 is the probability of the team picking a suboptimal action. $\psi_1 = \psi_2 + \psi_3$, where ψ_2 is the probability of no agent agreeing and the team picks a suboptimal action and ψ_3 is the probability of at least two agents agreeing and the team picks a suboptimal action. Hence, $p_{best}(m) = 1 - \psi_2 - \psi_3 = \tilde{p}_{best} - \psi_3 \geq \tilde{p}_{best} - \psi_4$, where ψ_4 is the probability of at least two agents agreeing. Let $\sigma^{max} = \max_{i_1, i_2} \sigma_{i_1, i_2}$, and i_1^* and i_2^* are the agents whose $\sigma_{i_1^*, i_2^*} = \sigma^{max}$. We have that $p_{best}(m) \geq \tilde{p}_{best} - \frac{n(n-1)}{2} \sigma^{max} \geq \tilde{p}_{best} - \frac{n(n-1)}{2} d_m^{(i_2^*)} p_{i_1^*, max} p_{i_2^*, max} \geq \tilde{p}_{best} - \frac{n(n-1)}{2} d_m^{(i_2^*)} \alpha p_{i_1^*, min} \alpha p_{i_2^*, min} \geq \tilde{p}_{best} - \frac{n(n-1)}{2} \alpha^2 \frac{1}{d_m^{(i_1^*)}}$ (as $p_{i, min} \leq \frac{1}{d_m}$). Hence, $p_{best}(m) \geq \tilde{p}_{best} - \frac{n(n-1)}{2} \alpha^2 \frac{1}{d_m^{min}} \rightsquigarrow \tilde{p}_{best} - p_{best}(m) \leq O(\frac{1}{d_m^{min}})$.

Now I show an upper bound: $p_{best}(m) = \tilde{p}_{best} - \psi_3 \leq \tilde{p}_{best} - \psi_5$, where ψ_5 is the probability of at least two agents agreeing and no agents vote for the optimal action. Let $\sigma^{min} = \min_{i_1, i_2} \sigma_{i_1, i_2}$; i_1^* and i_2^* are the agents whose $\sigma_{i_1^*, i_2^*} = \sigma^{min}$; and $p_{max, 0} = \max_{i \in \mathbf{T}} p_{i, 0}$. Without loss of generality, I assume that $d_m^{(i_2^*)} \geq d_m^{(i_1^*)}$. Hence, $p_{best}(m) \leq \tilde{p}_{best} - \frac{n(n-1)}{2} \sigma^{min} (1 - p_{max, 0})^{n-2} \leq \tilde{p}_{best} - \frac{n(n-1)}{2} d_m^{(i_1^*)} p_{i_1^*, min} p_{i_2^*, min} (1 - p_{max, 0})^{n-2} \leq \tilde{p}_{best} - \frac{n(n-1)}{2} d_m^{(i_1^*)} \frac{p_{i_1^*, max} p_{i_2^*, max}}{\alpha^2} (1 - p_{max, 0})^{n-2} \leq \tilde{p}_{best} - \frac{n(n-1)}{2} \alpha^{-2} \frac{1}{d_m^{i_2^*}} (1 - p_{max, 0})^{n-2} \leq \tilde{p}_{best} - \frac{n(n-1)}{2} \alpha^{-2} \frac{1}{d_m^{max}} (1 - p_{max, 0})^{n-2} \rightsquigarrow \tilde{p}_{best} - p_{best}(m) \geq \Omega(\frac{1}{d_m^{max}})$. ■

Hence, agents that change their d_m faster will converge faster to \tilde{p}_{best} . This is an important result when I consider later more complex scenarios where the d_m of the agents of the uniform team also change.

Note that \tilde{p}_{best} depends on the number of agents n (Equation 4.1). Now I show that the diverse team tends to always play the optimal action, as $n \rightarrow \infty$.

Theorem 4.1.3 \tilde{p}_{best} converges to 1, as $n \rightarrow \infty$. Furthermore, $1 - \tilde{p}_{best}$ converges exponentially to 0, that is, \exists constant c , such that $1 - \tilde{p}_{best} \leq c(1 - \frac{\epsilon}{2})^n$, $\forall n \geq \frac{2}{\epsilon}$. However, the performance of the uniform team improves as $n \rightarrow \infty$ only if $p_{s, 0} = \max_j p_{s, j}$, where s is the best agent.

Proof: By the previous proof, we know that when $m \rightarrow \infty$ the diverse team plays the optimal action with probability given by \tilde{p}_{best} . I show that $1 - \tilde{p}_{best} \rightarrow 0$ exponentially as $n \rightarrow \infty$ (this naturally induces $\tilde{p}_{best} \rightarrow 1$). I first compute an upper bound for $\sum_{i=1}^n (p_{i, 0} \prod_{j=1, j \neq i}^n (1 - p_{j, 0}))$: $\sum_{i=1}^n p_{i, 0} \prod_{j=1, j \neq i}^n (1 - p_{j, 0}) \leq \sum_{i=1}^n p_{i, 0} (1 - p_{min, 0})^{n-1} \leq n p_{max, 0} (1 - p_{min, 0})^{n-1} \leq n(1 - \epsilon)^{n-1}$ for $p_{max, 0} = \max_i p_{i, 0}$, $p_{min, 0} = \min_j p_{j, 0}$.

Since $\prod_{i=1}^n (1 - p_{i, 0}) \leq (1 - \epsilon)^n$, thus we have that $1 - \tilde{p}_{best} \leq (1 - \epsilon)^n + n(1 - \epsilon)^{n-1}$. So we only need to prove that there exists a constant c such that $(1 - \epsilon)^n + n(1 - \epsilon)^{n-1} \leq c(1 - \frac{\epsilon}{2})^n$, as follows: $\frac{(1 - \epsilon)^{n+1} + (n+1)(1 - \epsilon)^n}{(1 - \epsilon)^n + n(1 - \epsilon)^{n-1}} = (1 - \epsilon) \frac{1 - \epsilon + n + 1}{1 - \epsilon + n} = 1 - \epsilon + \frac{1 - \epsilon}{1 - \epsilon + n} \leq 1 - \frac{1}{2}\epsilon$, if $n \geq \frac{2}{\epsilon}$ (by

setting $\frac{1-\epsilon}{1-\epsilon+n} \leq \frac{\epsilon}{2}$). Hence, $\exists c$, such that $(1-\epsilon)^n + n(1-\epsilon)^{n-1} \leq c(1-\frac{\epsilon}{2})^n$ when $n \geq \frac{2}{\epsilon}$. Therefore, the performance converges exponentially.

For the uniform team, the probability of playing the action that has the highest probability in the pdf of the best agent converges to 1 as $n \rightarrow \infty$ [List and Goodin, 2001]. Therefore, the performance only increases as $n \rightarrow \infty$ if the optimal action is the one that has the highest probability. ■

Now I show that we can achieve further improvement in a diverse team by breaking ties in favor of the strongest agent.

Theorem 4.1.4 *When $m \rightarrow \infty$, breaking ties in favor of the strongest agent is the optimal tie-breaking rule for a diverse team.*

Proof: Let s be one of the agents. If we break ties in favor of s , the probability of voting for the optimal choice will be given by:

$$\tilde{p}_{best} = 1 - \prod_{i=1}^n (1 - p_{i,0}) - (1 - p_{s,0}) \left(\sum_{i=1, i \neq s}^n p_{i,0} \prod_{j=1, j \neq i, j \neq s}^n (1 - p_{j,0}) \right) \quad (4.2)$$

It is clear that Equation 4.2 is maximized by choosing agent s with the highest $p_{s,0}$. However, I still have to show that it is better to break ties in favor of the strongest agent than breaking ties randomly. That is, I have to show that Equation 4.2 is always higher than Equation 4.1.

Equation 4.2 differs from Equation 4.1 only on the last term. Therefore, I have to show that the last term of Equation 4.2 is smaller than the last term of Equation 4.1. Let's begin by rewriting the last term of Equation 4.1 as:

$$\frac{n-1}{n} \sum_{i=1}^n p_{i,0} \prod_{j=1, j \neq i}^n (1 - p_{j,0}) = \frac{n-1}{n} (1 - p_{s,0}) \sum_{i=1, i \neq s}^n p_{i,0} \prod_{j=1, j \neq i, j \neq s}^n (1 - p_{j,0}) + \frac{n-1}{n} p_{s,0} \prod_{j=1, j \neq s}^n (1 - p_{j,0})$$

This implies that:

$$\frac{n-1}{n} \sum_{i=1}^n p_{i,0} \prod_{j=1, j \neq i}^n (1 - p_{j,0}) \geq \frac{n-1}{n} (1 - p_{s,0}) \sum_{i=1, i \neq s}^n p_{i,0} \prod_{j=1, j \neq i, j \neq s}^n (1 - p_{j,0}).$$

We know that:

$$(1 - p_{s,0}) \sum_{i=1, i \neq s}^n p_{i,0} \prod_{j=1, j \neq i, j \neq s}^n (1 - p_{j,0}) = \frac{n-1}{n} (1 - p_{s,0}) \sum_{i=1, i \neq s}^n p_{i,0} \prod_{j=1, j \neq i, j \neq s}^n (1 - p_{j,0}) + \frac{1}{n} (1 - p_{s,0}) \sum_{i=1, i \neq s}^n p_{i,0} \prod_{j=1, j \neq i, j \neq s}^n (1 - p_{j,0})$$

Therefore, for the last term of Equation 4.2 to be smaller than the last term of Equation 4.1 I have to show that:

$$\frac{n-1}{n} p_{s,0} \prod_{j=1, j \neq s}^n (1 - p_{j,0}) \geq \frac{1}{n} (1 - p_{s,0}) \sum_{i=1, i \neq s}^n p_{i,0} \prod_{j=1, j \neq s, j \neq i}^n (1 - p_{j,0})$$

It follows that this equation will be true if:

$$p_{s,0} \geq (1-p_{s,0}) \frac{\sum_{i=1, i \neq s}^n p_{i,0} \prod_{j=1, j \neq i, j \neq s}^n (1-p_{j,0})}{(n-1) \prod_{j=1, j \neq s}^n (1-p_{j,0})} \rightsquigarrow p_{s,0} \geq (1-p_{s,0}) \frac{1}{n-1} \sum_{i=1, i \neq s}^n \frac{p_{i,0}}{(1-p_{i,0})} \rightsquigarrow \frac{p_{s,0}}{(1-p_{s,0})} \geq \frac{\sum_{i=1, i \neq s}^n \frac{p_{i,0}}{(1-p_{i,0})}}{n-1}$$

As s is the strongest agent the previous inequality is always true. This is because $\frac{p_{s,0}}{1-p_{s,0}} = \frac{\sum_{i=1, i \neq s}^n \frac{p_{i,0}}{(1-p_{i,0})}}{n-1}$ and $\frac{p_{s,0}}{1-p_{s,0}} \geq \frac{p_{i,0}}{(1-p_{i,0})} \forall i \neq s$. Therefore, it is always better to break ties in favor of the strongest agent than breaking ties randomly. ■

Next I show that with one additional assumption, not only the diverse team converges to \tilde{p}_{best} , but also p_{best} monotonically increases with m . My additional assumption is that higher utility actions have higher probabilities, i.e., if $U(a_j) \geq U(a_{j'})$, then $p_{i,j} \geq p_{i,j'}$.

Theorem 4.1.5 *The performance of a diverse team monotonically increases with m , if $U(a_j) \geq U(a_{j'})$ implies that $p_{i,j} \geq p_{i,j'}$.*

Proof: Let an event be the resulted choice set of actions of these n agents. I denote by $P(V)$ the probability of occurrence of any event in V (hence, $P(V) = \sum_{v \in V} p(v)$). I call it a winning event if in the event the action chosen by plurality is the best action a_0 (including ties). I assume that for all agents i , if $U(a_j) \geq U(a_{j'})$, then $p_{i,j} \geq p_{i,j'}$.

I show by mathematical induction that we can divide the probability of multiple suboptimal actions into a new action and $p_{best}(m+1) \geq p_{best}(m)$. Let λ be the number of actions whose probability is being divided. The base case holds trivially when $\lambda = 0$. That is, there is a new action, but all agents have a 0 probability of voting for that new action. In this case we have that p_{best} does not change, therefore $p_{best}(m+1) \geq p_{best}(m)$.

Now assume that we divided the probability of λ actions and it is true that $p_{best}(m+1) \geq p_{best}(m)$. I show that it is also true for $\lambda + 1$. Hence, let's pick one more action to divide the probability. Without loss of generality, assume it is action a_{d_m} , for agent c , and its probability is being divided into action $a_{d_{m+1}}$. Therefore, $p'_{c,d_m} = p_{c,d_m} - \beta$ and $p'_{c,d_{m+1}} = p_{c,d_{m+1}} + \beta$, for $0 \leq \beta \leq p_{c,d_m}$. Let $p_{best}^{after}(m+1)$ be the probability of voting for the best action after this new division, and $p_{best}^{before}(m+1)$ the probability before this new division. I show that $p_{best}^{after}(m+1) \geq p_{best}^{before}(m+1)$.

Let Γ be the set of all events where all agents voted, except for agent c (the order does not matter, so we can consider agent c is the last one to post its vote). If $\gamma \in \Gamma$ will be a winning event no matter if agent c votes for a_{d_m} or $a_{d_{m+1}}$, then changing agent c 's pdf will not affect the probability of these winning events. Hence, let $\Gamma' \subset \Gamma$ be the set of all events that will become a winning event depending if agent c does not vote for a_{d_m} or $a_{d_{m+1}}$. Given that $\gamma \in \Gamma'$ already happened, the probability of winning or losing is equal to the probability of agent c not voting for a_{d_m} or $a_{d_{m+1}}$.

Now let's divide Γ' in two exclusive subsets: $\Gamma_{d_{m+1}} \subset \Gamma'$, where for each $\gamma \in \Gamma_{d_{m+1}}$ action $a_{d_{m+1}}$ is in tie with action a_0 , so if agent c does not vote for $a_{d_{m+1}}$, γ will be a

winning event; $\Gamma_{d_m} \subset \Gamma'$, where for each $\gamma \in \Gamma_{d_m}$ action a_{d_m} is in tie with action a_0 , so if agent c does not votes for a_{d_m} , γ will be a winning event. I do not consider events where both a_{d_m+1} and a_{d_m} are in tie with a_0 , as in that case the probability of a winning event does not change (it is given by $1 - p'_{c,d_m} - p'_{c,d_m+1} = 1 - p_{c,d_m} - p_{c,d_m+1}$).

Note that for each $\gamma \in \Gamma_{d_m+1}$, the probability of a winning event equals $1 - p'_{c,d_m+1}$. Therefore, after changing the pdf of agent c , for each $\gamma \in \Gamma_{d_m+1}$, the probability of a wining event decreases by β . Similarly, for each $\gamma \in \Gamma_{d_m}$, the probability of a winning event equals $1 - p'_{c,d_m}$. Therefore, after changing the pdf of agent c , for each $\gamma \in \Gamma_{d_m}$, the probability of a winning event increases by β .

Therefore, $p_{best}^{after}(m+1) \geq p_{best}^{before}(m+1)$ if and only if $P(\Gamma_{d_m}) \geq P(\Gamma_{d_m+1})$. Note that $\forall \gamma \in \Gamma_{d_m+1}$ there are more agents that voted for a_{d_m+1} than for a_{d_m} . Also, $\forall \gamma \in \Gamma_{d_m}$ there are more agents that voted for a_{d_m} than for a_{d_m+1} . If, for all agents i , $p_{i,d_m} \geq p_{i,d_m+1}$, we have that $P(\Gamma_{d_m}) \geq P(\Gamma_{d_m+1})$. Therefore, $p_{best}^{after}(m+1) \geq p_{best}^{before}(m+1)$, so we still have that $p_{best}(m+1) \geq p_{best}(m)$. Also note that for the next step of the induction be valid, so that we can still divide the probability of one more action, it is necessary that $p'_{c,d_m} \geq p'_{c,d_m+1}$. ■

4.1.2 Generalizations

In the previous theorems I focused on the probability of playing the best action, assuming that $U(a_0) \gg U(a_j) \forall j \neq 0$. I show now that the theorems still hold in more general domains where r actions ($\mathbf{A}_r \subset \mathbf{A}$) have a significant high utility, i.e., $U(a_{j_1}) \gg U(a_{j_2}) \forall j_1 < r, j_2 \geq r$. Hence, I now focus on the probability of playing any action in \mathbf{A}_r . I assume that my assumptions are also generalized, i.e., $p_{i,j} > \epsilon \forall j < r$, and the number d_m of suboptimal actions ($a_j, j \geq r$) in the \mathbf{D}_m set increases with m for ST agents.

Theorem 4.1.6 *The previous theorems generalize to settings where $U(a_{j_1}) \gg U(a_{j_2}) \forall j_1 < r, j_2 \geq r$.*

Proof Sketch: I give here a proof sketch. We just have to generate new pdfs $p'_{i,j}$, such that $p'_{i,0} = \sum_{j=0}^{r-1} p_{i,j}$, and $p'_{i,b} = p_{i,b+r-1}, \forall b \neq 0$. We can then reapply the proofs of the previous theorems, but replacing $p_{i,j}$ by $p'_{i,j}$. Note that this does not guarantee that all agents will tend to agree on the same action in \mathbf{A}_r ; but the team will still tend to pick any action in \mathbf{A}_r , since the agents are more likely to agree on actions in \mathbf{A}_r than on actions in $\mathbf{A} \setminus \mathbf{A}_r$. ■

Now I discuss a different generalization: what happens when $p_{i,0}$ decreases as m increases (\forall agents i). If $p_{i,0} \rightarrow \tilde{p}_{i,0}$ as $m \rightarrow \infty$, the performance in the limit for a diverse team will be \tilde{p}_{best} evaluated at $\tilde{p}_{i,0}$. Moreover, even if $p_{i,0} \rightarrow 0$, my conclusions about

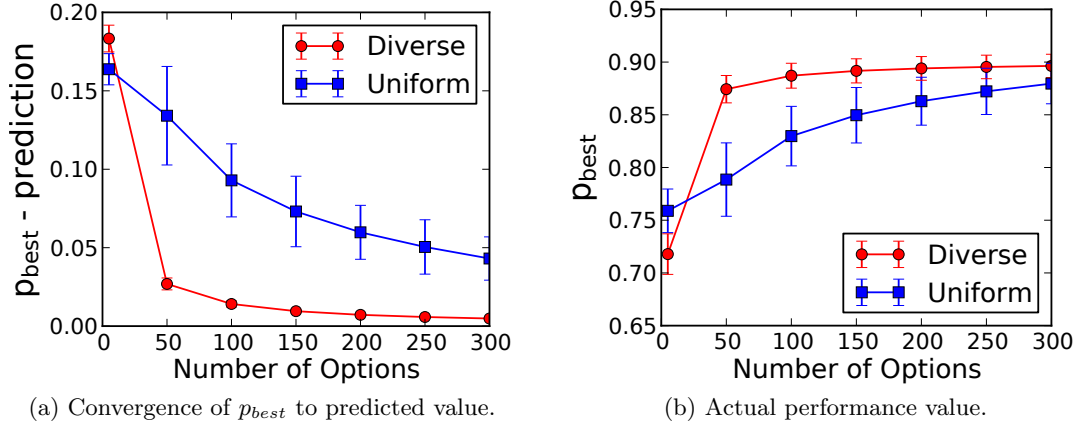


Figure 4.1: Comparing diverse and uniform when uniform also increases d_m .

relative team performance are not affected as long as we are comparing two ST teams that have similar $p_{i,0}$: the same argument as in Corollary 1 implies that the team with faster growing d_m will perform better.

4.2 Experimental Analysis

4.2.1 Synthetic Experiments

I present synthetic experiments, in order to better understand what happens in real systems. I generate agents by randomly creating pdfs and calculate the probability of playing the best action (p_{best}) of the generated teams. I use a uniform distribution to generate all random numbers. When creating a pdf, I rescale the values assigned randomly, so that the overall sum of the pdf is equal to 1.

As I said earlier, uniform teams composed by NST agents is an idealization. In more complex domains, the best agent will not behave exactly like an NST agent; the number of suboptimal actions with a non-zero probability (d_m) will also increase as the action space gets larger. I perform synthetic experiments to study this situation. I consider that the best agent is still closer to an NST agent, therefore it increases its d_m at a slower rate than the agents of the diverse team.

In my first experiment, I use teams of 4 agents. For each agent of the diverse team, $p_{i,0}$ is chosen randomly between 0.6 and 0.7. The remaining is distributed randomly from 10% to 20% of the next best actions (the number of actions that will receive a positive probability is also decided randomly). For the uniform team, I make copies of the best agent (with highest $p_{i,0}$) of the diverse team, but distribute the remaining probability randomly from 1% to 3% of the next best actions.

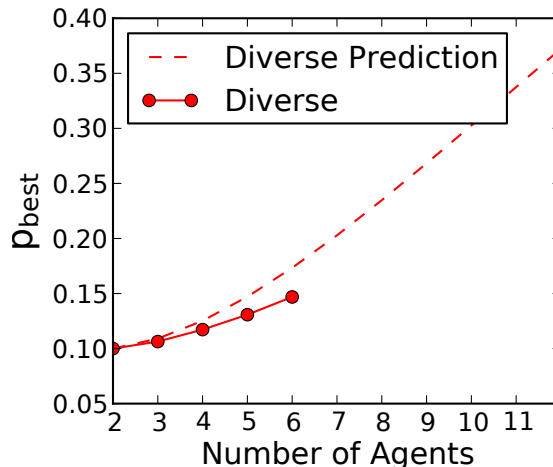


Figure 4.2: p_{best} of a diverse team as the number of agents increases.

We can see the average result for 200 random teams in Figure 4.1, where in Figure 4.1(a) I show the difference between the performance in the limit (\tilde{p}_{best}) and the actual $p_{best}(m)$ for the diverse and the uniform teams; in Figure 4.1(b) I show the average $p_{best}(m)$ of the teams. As can be seen, when the best agents increase their d_m at a slower rate than the agents of the diverse team, the uniform teams converge slower to \tilde{p}_{best} . Even though they play better than the diverse teams for a small m , they are surpassed by the diverse teams as m increases. However, because \tilde{p}_{best} of the uniform teams is actually higher than the one of the diverse teams, eventually the performance of the uniform teams get closer to the performance of the diverse teams, and will be better than the one of the diverse teams again for a large enough m .

This situation is expected according to Theorem 4.1.1. If the d_m of the best agent also increases as m gets larger, the uniform team will actually behave like a diverse team and also converge to \tilde{p}_{best} . $\tilde{p}_{best}^{uniform} \geq \tilde{p}_{best}^{diverse}$, as the best agent has a higher probability of playing the optimal action. Hence, in the limit the uniform team will play better than the diverse team. However, as we saw in Corollary 4.1.2, the speed of convergence is in the order of $1/d_m$. Therefore, the diverse team will converge faster, and can overcome the uniform team for moderately large m .

As Theorem 4.1.3 only holds when $m \rightarrow \infty$, I also explore the effect of increasing the number of agents for a large m . The \tilde{p}_{best} of a team of agents is shown as the dashed line in Figure 4.2. I am plotting for agents that have a probability of playing the best action of only 10%, but as we can see the probability quickly grows as the number of agents increases. I also calculate p_{best} for random teams from 2 to 6 agents (shown as the continuous line), when there are 300 available actions. Each agent has a probability of playing the best action of 10%, and the remaining probability is randomly distributed

over the 10% next best actions. As can be seen, the teams have a close performance to the expected. I only show up to 6 agents because it is too computationally expensive to calculate the pdfs of larger teams.

4.2.2 Computer Go

I present now results in a real system. I use in my experiments 4 different Go software: Fuego 1.1, GnuGo 3.8, Pachi 9.01, MoGo 4, and two (weaker) variants of Fuego (Fuego Δ and Fuego Θ), in a total of 6 different, publicly available, agents. Fuego is considered the strongest agent among all of them. Fuego is an implementation of the UCT Monte Carlo Go algorithm, therefore it uses heuristics to simulate games in order to evaluate board configurations. Fuego uses mainly 5 heuristics during these simulations, and they are executed in a hierarchical order. The original Fuego agent follows the order <Atari Capture, Atari Defend, Lowlib, Pattern> (The heuristic called Nakade is not enabled by default). My variation called Fuego Δ follows the order <Atari Defend, Atari Capture, Pattern, Nakade, Lowlib>, while Fuego Θ follows the order <Atari Defend, Nakade, Pattern, Atari Capture, Lowlib>. Also, Fuego Δ and Fuego Θ have half of the memory available when compared with the original Fuego.

All my results are obtained by playing either 1000 games (to evaluate individual agents) or 2000 games (to evaluate teams), in a HP dl165 with dual dodeca core, 2.33GHz processors and 48GB of RAM. I compare results obtained by playing against a fixed opponent. Therefore, I evaluate systems playing as white, against the original Fuego playing as black. I removed all databases and specific board size knowledge of the agents, including the opponent. I call Diverse as the team composed of all 6 agents, and Uniform as the team composed of 6 copies of Fuego. Each agent is initialized with a different random seed, therefore they will not vote for the same action all the time in a given world state, due to the characteristics of the search algorithms. In all the graphs I present in this section, the error bars show the confidence interval, with 99% of confidence ($p = 0.01$).

I evaluate the performance of the teams over 7 different board sizes. I changed the time settings of individual agents as I increased the board size, in order to keep their strength as constant as possible. The average winning rates of the team members is shown in Table 4.2, while Table 4.3 show the winning rates of the individual agents¹.

We can see my results in Figure 4.3 (a). Diverse improves from 58.1% on 9x9 to 72.1% on 21x21, an increase in winning rate that is statistically significant with $p < 2.2 \times 10^{-16}$. This result is expected according to Theorem 4.1.1. Uniform changes from 61.0% to

¹In my first experiment, Diverse improved from 56.1% on 9x9 to 85.9% on 19x19. I noted, however, that some of the diverse agents were getting stronger in relation to the opponent as the board size increased. Hence, by changing the time setting to keep the strength constant, I am actually making my claims harder to show, not easier.

Team	9x9	11x11	13x13	15x15	17x17	19x19	21x21
Diverse	32.2%	30.8%	29.6%	29.4%	31.5%	31.9%	30.3%
Uniform	48.1%	48.6%	46.1%	48.0%	49.3%	46.9%	46.6%

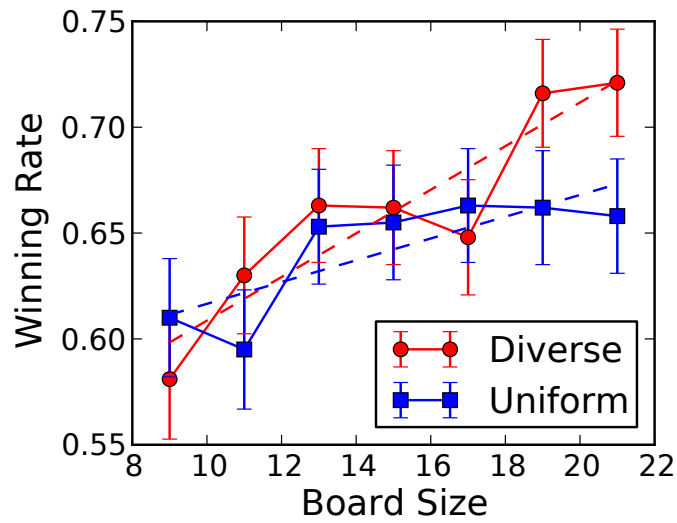
Table 4.2: Average winning rates of the team members across different board sizes. Note that these are not the winning rates of the teams.

Agent	9x9	11x11	13x13	15x15	17x17	19x19	21x21
Fuego	48.1%	48.6%	46.1%	48.0%	49.3%	46.9%	46.6%
GnuGo	1.1%	1.1%	1.9%	1.9%	4.5%	6.8%	6.1%
Pachi	25.7%	22.9%	25.8%	26.9%	23.5%	20.8%	11.0%
MoGo	27.6%	26.4%	22.7%	22.0%	27.1%	30.1%	27.1%
Fuego Δ	45.7%	45.8%	42.2%	40.4%	43.0%	44.5%	47.4%
Fuego Θ	45.5%	40.2%	39.2%	37.6%	41.8%	42.3%	43.6%

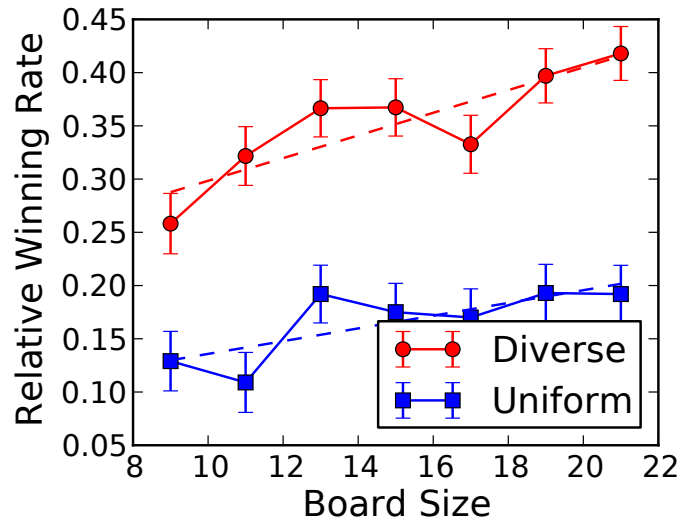
Table 4.3: Winning rates of each one of the agents across different board sizes.

65.8%, a statistically significant improvement with $p = 0.0018$. As we saw before, an increase in the performance of Uniform can also be expected, as the best agent might not be a perfect *NST* agent. A linear regression of the results of both teams gives a slope of 0.010 for the diverse team (adjusted R^2 : 0.808, $p = 0.0036$) and 0.005 for the uniform team (adjusted R^2 : 0.5695, $p = 0.0305$). Therefore, the diverse team improves its winning rate faster than the uniform team. To check if this is a significant difference, I evaluate the interaction term in a linear regression with multiple variables. We find that the influence of board size is higher on Diverse than on Uniform with $p = 0.0797$ (estimated coefficient of “size of the board \times group type”: -10.321 , adjusted R^2 : 0.7437). Moreover, on the 9x9 board Diverse is worse than Uniform ($p = 0.0663$), while on the 21x21 board Diverse is better with high statistical significance ($p = 1.941 \times 10^{-5}$). I also analyze the performance of the teams subtracted by the average strength of their members (Figure 4.3 (b)), in order to calculate the increase in winning rate achieved by “teamwork” and compensate fluctuations on the winning rate of the agents as we change the board size. Again, the diverse team improves faster than the uniform team. A linear regression results in a slope of 0.0104 for Diverse (adjusted R^2 : 0.5549, $p = 0.0546$) and 0.0043 for Uniform (adjusted R^2 : 0.1283, $p = 0.258$).

I also evaluate the performance of teams of 4 agents (Diverse 4 and Uniform 4). For Diverse 4, I removed Fuego Δ and Fuego Θ from the Diverse team. As can be seen in Figure 4.4, the impact of adding more agents is higher for the diverse team in a larger board size (21x21). In the 9x9 board, the difference between Diverse 4 and Diverse 6 is only 4.4%; while in 21x21 it is 14%. Moreover, we can see a higher impact of adding



(a) Absolute winning rates.



(b) Relative to the average strength of team members.

Figure 4.3: Winning rate in the real Computer Go system.

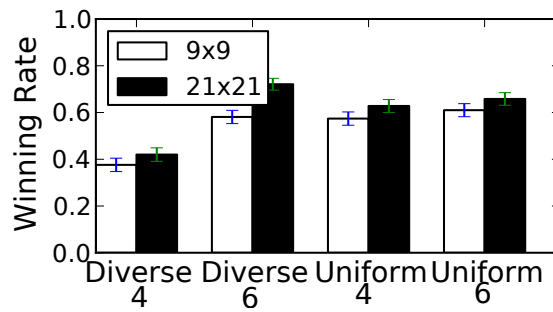


Figure 4.4: Winning rates for 4 and 6 agents teams.

agents for the diverse team, than for the uniform team. These results would be expected according to Theorem 4.1.3.

As can be seen, the predictions of my theory holds: the diverse team improves significantly as I increase the action space. The improvement is enough to make it change from playing worse than the uniform team on 9x9 to playing better than the uniform team with statistical significance on the 21x21 board. Furthermore, I show a higher impact of adding more agents when the size of the board is larger.

4.2.3 Analysis

To test the assumptions of my model, I estimate a pdf for each one of the agents. For each board size, and for each one of 1000 games from my experiments, I randomly choose a board state between the first and the last movement. I make Fuego evaluate the chosen board, but I give it a time limit 50x higher than the default one. Therefore, I use this much stronger version of Fuego to approximate the true ranking of all actions. For each board size, I run all agents in each board sample and check in which position of the approximated true ranking they play. This allows me to build a histogram for each agent and board size combination. Some examples can be seen in Figure 4.5. We can see that a strong agent, like Fuego, has most of its probability mass on the higher ranked actions, while weaker agents, like GnuGo, has the mass of its pdf distributed over a larger set of actions, creating a larger tail. Moreover, the probability mass of GnuGo is spread over a larger number of actions when I increase the size of the board.

I study how the pdfs of the agents change as we increase the action space. My hypothesis is that weaker agents will have a behavior closer to *ST* agents, while stronger agents to *NST* agents. In Figure 4.6 (a) I show how many actions receive a probability higher than 0. As can be seen, Fuego does not behave exactly like an *NST* agent. However, it does have a slower growth rate than the other agents. A linear regression gives the following slopes: 13.08, 19.82, 19.05, 15.82, 15.69, 16.03 for Fuego, Gnugo, Pachi, Mogo, Fuego Δ and Fuego Θ , respectively (R^2 : 0.95, 0.98, 0.94, 0.98, 0.98, 0.98, respectively). It is clear, therefore, that the probability mass of weak agents is distributed into bigger sets of actions as we increase the action space, and even though the strongest agent does not behave in the idealized way it does have a slower growth rate.

I also verify how the probability of playing the best action changes for each one of the agents as the number of actions increase. Figure 4.6 (b) shows that even though all agents experience a decrease in $p_{i,0}$, it does not decrease much. From 9x9, all the way to 21x21, I measure the following decrease: 20%, 23%, 39%, 26%, 28%, 22%, for Fuego, Gnugo, Pachi, Mogo, Fuego Δ and Fuego Θ , respectively. Hence, on average, they decreased about 25% from 9x9 to 21x21. Even though my assumption about $p_{i,0}$ does not hold perfectly, the

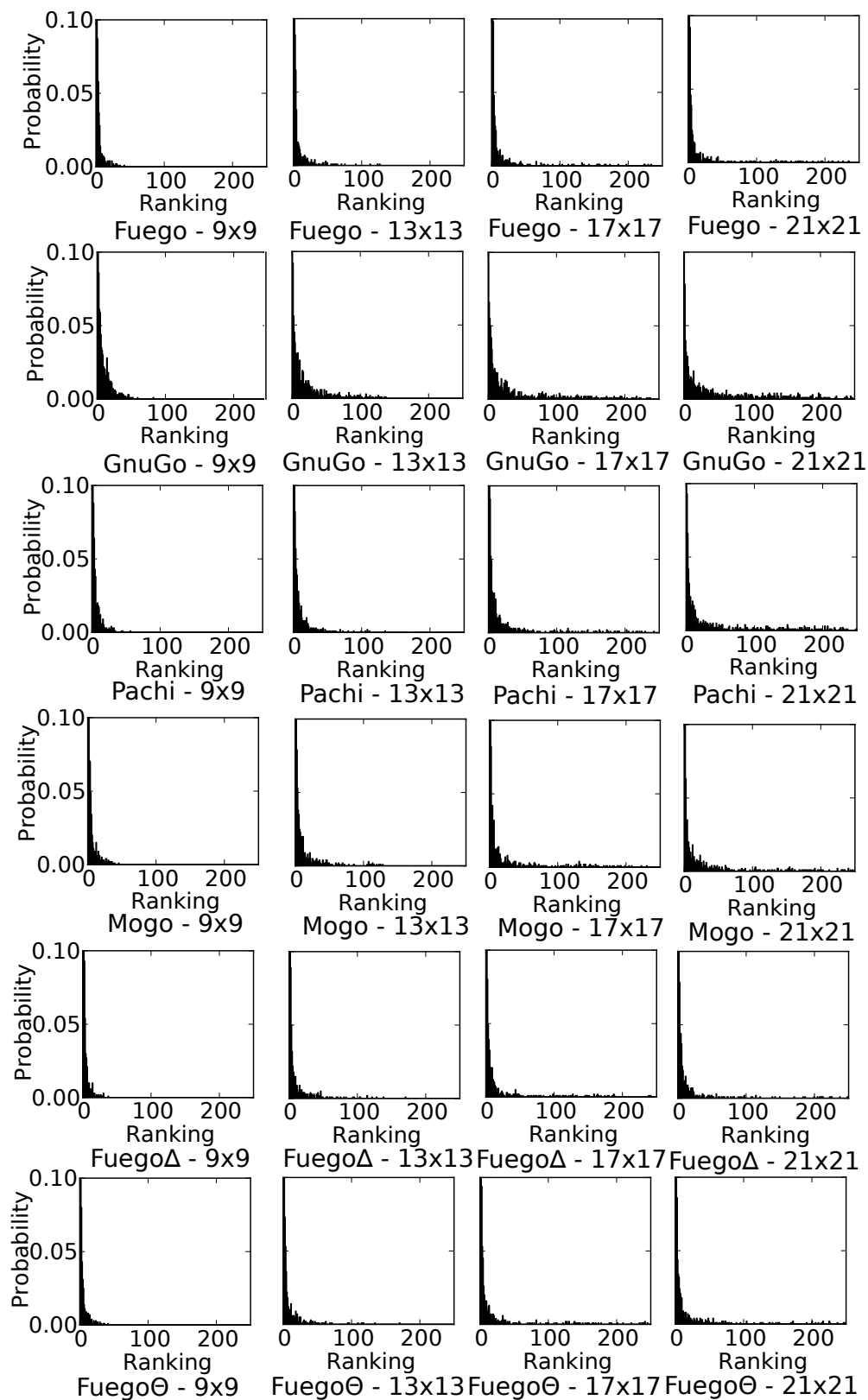


Figure 4.5: Histograms of agents for different board sizes.

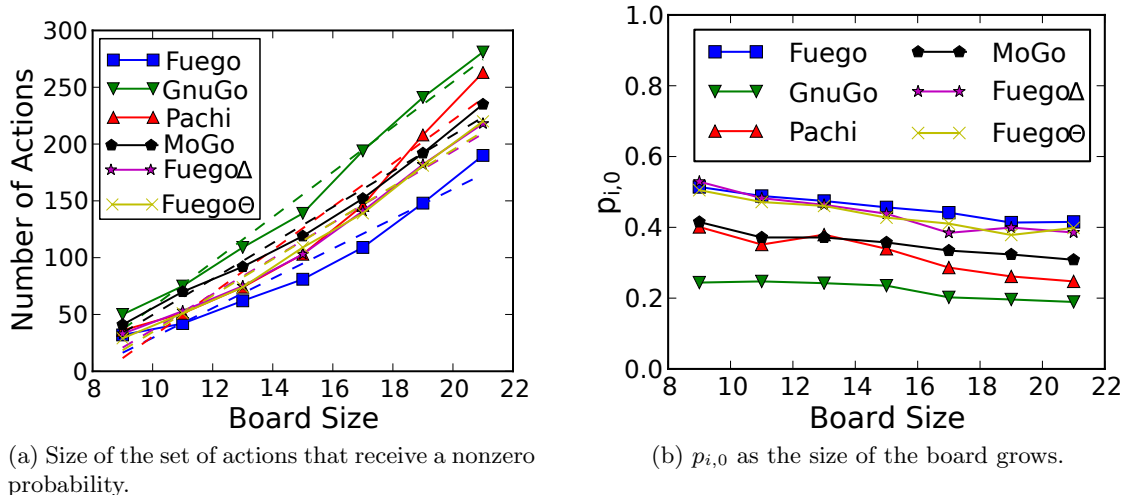


Figure 4.6: Verifying the assumptions in the real system.

predictions of my model are still verified. Therefore, the amount of decrease experienced is not enough to avoid that the diverse team increases in performance as the action space grows.

4.3 Conclusion and Discussions

Diversity is an important point to consider when forming teams. In this chapter I present a new model that captures better than previous ones the intuitive notion of diverse agents as agents that tend to disagree. This model allows me to make new predictions. I show that the performance of diverse teams increases as the size of the action space gets larger. Uniform teams may also increase in performance, but at a slower pace than diverse teams. Therefore, even though a diverse team may start playing worse than a uniform team, it can eventually outperform the uniform team as the action space increases. Besides, I show that in large action spaces the performance of a diverse team converges exponentially fast to the optimal one as the number of agents increases.

I start my model with the notion of *spreading tail* (ST) and *non-spreading tail* (NST) agents. ST agents are agents that have a non-zero probability over a larger set of actions as the action space increases, while NST agents always have a constant number of actions with non-zero probability. I define a diverse team as a team of ST agents, and a uniform team as a team of NST agents. Therefore, my focus change from modeling diverse teams as teams with different agents (as in Chapter 3), to focusing on diverse teams as teams where the agents tend to disagree. This change allows me to make new predictions that were not possible before.

Note that my model does *not* say that an *NST* agent will never vote for a new action. I define the pdfs of the agents by the rankings of the actions. Hence, when the number of actions increases from a certain number x_0 to a new number x_1 , a new action a^* may be the action with highest utility. Therefore, an agent will assign to a^* the same probability that it assigned before to the previously best action when the number of actions was only x_0 . A uniform team made of copies of the best agent also does *not* mean that the agents always vote for the same actions. The vote of each agent is a sample from a pdf, so copies of a single agent may or may not vote for the same action. In fact, we observe an increase in performance by voting among multiple copies of a single agent, both theoretically and experimentally.

The division of agents into two types (*ST* and *NST*) is, however, only an idealization, that allows me to isolate and study in detail the effect of diversity. A very strong agent will normally assign most of its probability mass to the actions with the highest utility, so in the extreme its pdf would never change by adding new actions. In reality, however, it may also consider a larger set of actions as the action space grows. Therefore, I relax my model, and introduce the hypothesis that the best agent spreads the tail of its pdf at a slower pace than weaker agents. I show that because of this effect, a diverse team increases in performance faster than uniform teams, and I illustrate this phenomenon with synthetic experiments. Hence, even in a relaxed model where both diverse and uniform teams are composed of *ST* agents, a diverse team still outperforms a uniform team as the action space grows. The effect, however, is transient, as a uniform team may still have a higher convergence point than a diverse team, so in extreme large action spaces it would again outperform the diverse team. If the agents have the same probability of playing the best action, however, then it is clear that in the limit the diverse team will always be better than the uniform team.

My model needs one strong assumption: that the probability of the individual agents voting for the best action does not change as the action space increases. This assumption allows my analysis to be cleaner, although it may not hold perfectly in a real system. In fact, in my Computer Go experiments we did observe a decrease in the probability of the agents voting for the best action. However, even though the assumption did not hold perfectly, the predictions of my theory holds: a diverse team significantly increased in performance as the action space got larger. Clearly, a decrease in the probability of the individual agents voting for the best action will decrease the performance of a team, while the effects studied in this paper will increase the performance. Therefore, as long as the decrease is not large enough to counter-balance the effect under study, we are still going to observe an increase in performance as the action space gets larger. Moreover, as I discuss in my generalizations, the argument that teams that spread the tail faster

converge faster is still valid when the assumption does not hold; hence if the agents are equally strong (i.e., the individual agents have the same probability of voting for the best action) the team with faster growing tail will always perform better.

As mentioned, I verified my theory in a real system of Computer Go playing agents. Not only a real diverse team of agents effectively increased in performance as the board size increased, but I also verified that the strongest agent indeed spreads the tail of its pdf at a slower rate than other weaker agents. I also verified that both diverse and uniform teams increase in performance, but the diverse team increased two times faster. This is explained by the relaxed version of my model, when I predict diverse teams to converge faster than uniform teams, as illustrated by my synthetic experiments.

In the next chapter, I will study diverse and uniform teams in the context of design problems, where the number of optimal solutions must be maximized, allowing a human to choose according to aesthetics or compromises that cannot be formalized.

Chapter 5

SO MANY OPTIONS, BUT WE NEED THEM ALL

*Creativity is allowing yourself to make mistakes.
Art is knowing which ones to keep.
(Scott Adams)*

5.1 Introduction

Teams of voting agents are a powerful tool for finding the optimal solution in many applications [Mao et al., 2013, Bachrach et al., 2012, Soejima et al., 2010, Polikar, 2012, Isa et al., 2010]. Voting is a popular approach since it is easily parallelizable, it allows the re-use of existing agents, and there are theoretical guarantees for finding one optimal choice [Conitzer and Sandholm, 2005, List and Goodin, 2001].

For design problems, however, finding one optimal solution is not enough. For example, it could be mathematically optimal under measurable metrics but lack aesthetic qualities or social acceptance by the target public. Besides, the solution could have a poor performance in some key objective of a multi-objective optimization problem. Essentially, designers need to explore a large set of optimal alternatives, to pick one solution not only according to her aesthetic taste (and/or the one of the target public), but also according to preferences that may be unknown or not formalized, especially when there are multiple optimization objectives leading to intricate trade-offs and compromises [Gerber, 2007, Woodbury and Burrow, 2006, Radford and Gero, 1980, van Langen and Brazier, 2006, Gero and Sosa, 2008].

Hence, we actually need systems that find as many optimal solutions as possible, allowing a human to explore such optimal alternatives to make a choice. Even if a user does not want to consider too many solutions, they can be filtered and clustered [Erhan et al., 2014], and be presented in manageable ways [Smith et al., 2010], allowing her to

easily make an informed choice. Therefore, a system of voting agents that produces a unique optimal solution is insufficient, and I propose the novel social choice problem of maximizing the number of optimal alternatives found by a voting system. As ranked voting may suffer from noisy rankings when using existing agents (Chapter 6), I study multiple plurality voting iterations, allowing great applicability and re-use of existing agents.

Traditionally, social choice studies the optimality of voting rules, assuming a certain noise models for the agents, and rankings composed of a linear order over alternatives [Conitzer and Sandholm, 2005, Caragiannis et al., 2013, List and Goodin, 2001]. Hence, there is a single optimal choice, and a system is successful if it can return that optimal choice with high probability. More recently, several works have been considering cases where there is a partial order over alternatives [Xia and Conitzer, 2011a, Procaccia et al., 2012], or where the agents output pairwise comparisons instead of rankings [Elkind and Shah, 2014]. However, these works still focus on finding an optimal alternative, or a fixed-sized set of optimal alternatives (where the size is known beforehand). Therefore, they still provide no help in finding the maximum set of optimal solutions. Moreover, they assume agents that are able to output comparisons among all actions with fairly good precision, and the use of multiple voting iterations has never been studied. When considering agents with different preferences, the field is focused on verifying if voting rules satisfy a set of axioms that are considered to be important to achieve fairness [Nurmi, 1987].

Meanwhile, the computational design literature has not yet found the potential of teams of voting agents. They mainly study traditional optimization techniques to find optimal solutions, such as genetic algorithms [Miles et al., 2001, Yi and Malkawi, 2009, Gerber and Lin, 2013], particle swarm optimization [Luh et al., 2011, Felkner et al., 2013], ant colony optimization [Luh and Lin, 2009], immune systems [Zhao et al., 2014], etc. Many researchers also explore the potential of swarms of agents that interact on the geometric space to emerge aesthetically complex shapes (but lacking any optimization attempt) [Snooks, 2011, Vehlken, 2014, Aranda and Lasch, 2006, Baharlou and Menges, 2013, Ireland, 2009, Carranza and Coates, 2000, Hanna, 2005]. Therefore, the literature still lacks a deep study of other multi-agent techniques and ideas.

Hence, in this chapter I bring together the social choice and computational design fields, offering new perspectives to both literatures. I present a theoretical study of which kinds of teams are desirable for design problems, and how their size may effect optimality. In doing so, I show many novel results for the study of multi-agent systems. For example, instead of studying agents with different preferences in order to verify fairness axioms, as in traditional social choice, I show here that agents with different preferences are actually

fundamental when voting to find a “truth” (i.e., optimal decisions). On the other hand, agents with the same set of preferences significantly harm the performance, and in general the number of optimal solutions *decreases* as the size of the team grows. Such results were never seen before in the social choice literature, as voting to estimate optimality is normally studied separated from voting as a way to aggregate agents with different preferences.

My theoretical development draws a novel connection between social choice and *number theory*, instead of the traditional connections with *bayesian probability theory*. This novel connection allows me to show, for instance, that the optimal diverse team size is constant with high probability, and a prime number of optimal actions may impose problems. I also show that we can maximize the number of optimal solutions with agents with different preferences as the team size grows, as long as the team size grows *carefully*.

Moreover, I simulate design agents in synthetic experiments to further study my model, confirming the predictions of my theory and providing realistic insights into what happens when systems run with bounded computational time. Finally, I present experiments in a highly relevant domain: architectural design, where I show teams of real design agents that vote to choose the best qualifying and energy-efficient design solutions for buildings. Such domain is fundamental in the current scenario where we must find energy-efficient solutions for our modern life-style, since it is known that the early design of a building has a major impact in its performance through-out its whole life-span [Lin and Gerber, 2014, Bogenstätter, 2000, Echenagucia et al., 2015]. I study actual teams of voting agents, and show that by aggregating their opinions, we are able to find a large percentage of optimal solutions, significantly overcoming single agents. I also discuss how my model can explain and make predictions about the real system, and how it can guide researchers in computational design when developing novel agents for design.

5.2 Design Domains

I consider in this chapter domains where the objective is to find the highest number of optimal solutions. I show that design is one of such domains. With the rapid development of computation, algorithmic techniques have been emerging as an important approach in design [Terzidis, 2006]. For example, Gero [2000] argues that computational approaches can be used to increase the space of design exploration and the creativity in designs.

One of the most common computational design approaches is to use *parametric designs* [Vierlinger and Bollinger, 2014, Globa et al., 2014, Erhan et al., 2014], where a human designer creates an initial design of a product using *computer-aided design* tools. However, instead of manually deciding all aspects of the product, she leaves *free parameters*, whose

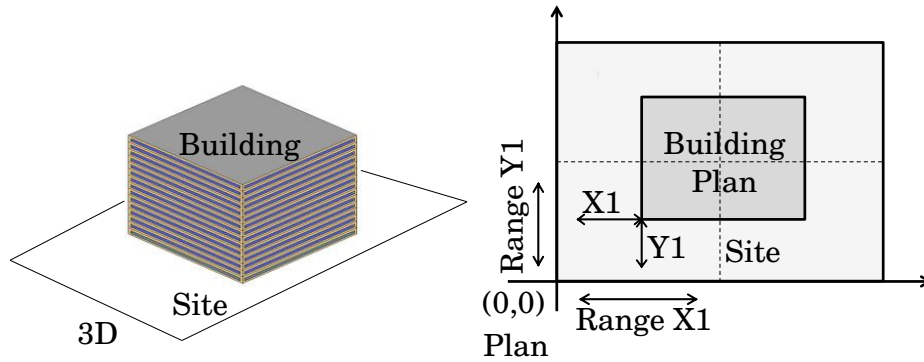


Figure 5.1: A parametric design of a building, showing two parameters: $X1$ and $Y1$.

values can be modified to change the design of the product. It is up to the designer to decide which parameters are going to be available, their valid types and their valid range.

This approach is used because design is an inherently complex problem [Simon, 1973]. Although a human is able to test and evaluate a few solutions looking for optimality, the number of different possibilities that she can manually create is highly limited, especially under the (common) hard time-constraints. In Figure 5.1, I show a simple example in the context of architectural design, where the parameters $X1$ and $Y1$ are being used to specify the position of the lower left corner of the building relative to the site boundary.

The design of a product normally occurs over multiple phases, where increasing levels of details are decided and optimized. My work is focused on the initial design phase, when multiple possible design alternatives are analyzed in order to choose one for further study and optimization. This initial design phase is, however, very important to the final performance of a product [Smith et al., 2010, Holzer et al., 2007]. For example, in the context of architectural design (as how I explore later in my experiments), it has been acknowledged that it has a high impact on the overall building performance [Bogenstätter, 2000, Lin and Gerber, 2014, Echenagucia et al., 2015, Yi and Malkawi, 2009].

Design problems are in general multi-objective [Lin and Gerber, 2012, Keough and Benjamin, 2010], since a product normally must be optimized across different objectives. For example, a product should have a low cost, but at the same time high quality, two highly-contradictory objectives. Hence, there are a large number of optimal solutions, all tied in a *Pareto frontier*. For the computational system, these optimal solutions are all equivalent. However, a human may have unknown preferences, may dynamically decide to value some objective over another when handling intricate trade-offs, and/or may choose the option that most pleases her own aesthetic taste or the one of the target public/client.

Note that choosing a design according to aesthetics is an undefined problem, since there are no formal definitions to compare among different options. Hence, the best that a

system can do is to provide a human with a large number of optimal solutions (according to other measurable factors), allowing her to freely decide among equally optimal solutions — but most probably with different aesthetic qualities.

Therefore, it is natural that in design problems we are going to have many possible solutions, and we want to find as many optimal ones as possible. In fact, the exploration of a large space of possible alternatives is essential in design, as recently shown by Woodbury and Burrow [2006], van Langen and Brazier [2006], and Gero and Sosa [2008]. There are many benefits in discovering a large number of optimal solutions, and I list some of them below:

Knowledge “Does not Hurt”: I argue that having more optimal solutions to choose from is not worse than having less. Although some works in psychology show that humans may get frustrated in the face of too many options, especially under time pressure [Haynes, 2009, Iyengar and Lepper, 2000], I argue that if a designer has enough time or motivation to analyze only x solutions, she can do so with a system that provides more than x optimal solutions by sampling the exact amount that she desires. However, she will never be able to do so with a system that provides less than x optimal solutions.

Note also that the works in psychology [Haynes, 2009, Iyengar and Lepper, 2000] were taken in the context of consumers deciding among products to purchase, not in the context of design exploration. As mentioned before, in design the necessity of large exploration spaces is recognized [Gerber, 2007, Woodbury and Burrow, 2006, van Langen and Brazier, 2006, Gero and Sosa, 2008].

Moreover, as I discuss in detail later, voting systems could be combined with another system that identify and eliminate solutions that are similar by applying clustering and analysis techniques, and that presents the optimal alternatives to a human in a manageable way [Erhan et al., 2014, Smith et al., 2010], so that every solution that the human looks at is meaningful.

Knowledge Increases Confidence in Optimality: In general design problems, the true Pareto frontier is unknown. Genetic algorithms are widely used in order to *estimate* it. The only knowledge available for the system to evaluate the optimality is in comparison with the other solutions that are also being evaluated during the optimization process [Lin and Gerber, 2014]. Many apparently “optimal” solutions are actually discovered to be sub-optimal as we find more solutions. Hence, finding a higher number of optimal solutions decreases the risk of a designer picking a wrong choice that was initially outputted as “optimal” by a system (for example, the single agents, as I will show later).

Knowledge Increases Aesthetic Qualities: If a human has a larger set of optimal solutions to choose from, there is a greater likelihood that at least one of these solutions

is going to be of high aesthetic quality according to her preferences, or the ones of the target public [Gero and Sosa, 2008].

Knowledge Increases Diversity of Options: In general, when a system x has more optimal solutions available than a system y , it does not necessarily imply that the solutions in the system x are more similar, while the optimal solutions in y are more different/diverse. In fact, all things equal (i.e., the algorithms are equally able to find unique solutions), the greater the amount of optimal solutions, the higher the likelihood that we have more diverse solutions available.

Of course we could have some algorithm x that produces many optimal solutions by creating small variations of one unique solution, but here I do not consider these potentially misleading systems. Again, I assume that such solutions could be identified and filtered by another system [Erhan et al., 2014, Smith et al., 2010].

5.3 Agent Teams for Design Problems

I present my theory of agent teams for design problems. I consider teams that vote together at each possible decision point of the design of a product (for example, they may vote for the value of each parameter, in a parametric design). Before showing my theoretical development, I first show an example to give an intuitive idea about my results. Then, in Section 5.3.2 and 5.3.3 I show my formal results.

5.3.1 Example

Let's consider a parametric design problem with two different parameters: ω_1 and ω_2 . Let's say that for ω_1 there are two possible optimal values: a_0 or a_1 . For ω_2 , let's assume three possible optimal values: a_1 , a_2 , a_3 . I consider that any solution with an optimal value for each parameter is optimal, and hence there are 6 (and only 6) possible optimal vectors: $\langle a_0, a_1 \rangle$, $\langle a_0, a_2 \rangle$, $\langle a_0, a_3 \rangle$, $\langle a_1, a_1 \rangle$, $\langle a_1, a_2 \rangle$, $\langle a_1, a_3 \rangle$.

Of course the information of which actions are optimal is not known before-hand, otherwise we would not need a design system at all. I list here the optimal values only for the purpose of my example.

I will study teams where each agent outputs a value for each parameter, and these opinions are aggregated by voting. Voting across each parameter, however, will only produce a single solution. Therefore, I consider multiple voting iterations (where one iteration goes across all parameters). At each iteration, we may find a new optimal solution, we may repeat a solution that was already found, or we may fail to find an optimal solution altogether.

Solution	$\langle a_0, a_1 \rangle$	$\langle a_0, a_2 \rangle$	$\langle a_0, a_3 \rangle$	$\langle a_1, a_1 \rangle$	$\langle a_1, a_2 \rangle$	$\langle a_1, a_3 \rangle$
6 Agents	0.0160	0.1310	0.0160	0.0822	0.6724	0.0822
12 Agents	0.0033	0.0715	0.0033	0.0393	0.8430	0.0393

(a) Uniform Team

Solution	$\langle a_0, a_1 \rangle$	$\langle a_0, a_2 \rangle$	$\langle a_0, a_3 \rangle$	$\langle a_1, a_1 \rangle$	$\langle a_1, a_2 \rangle$	$\langle a_1, a_3 \rangle$
6 Agents	0.1590	0.1590	0.1590	0.1590	0.1590	0.1590
12 Agents	0.1638	0.1638	0.1638	0.1638	0.1638	0.1638

(b) Diverse Team

Table 5.2: Probability of outputting each possible optimal solution, for the diverse and uniform teams.

plurality voting in Table 5.2 (b). For a 6 agents team I consider one copy of each agent, and for a 12 agents team I consider two copies of each agent. As we can notice, even though now the probabilities of voting for optimal solutions sum up to $0.954 \leq 1$ with 6 agents, they are evenly distributed over all optimal solutions. Hence, if we run 10 voting iterations, each option is expected to be seen 1.5 times. Moreover, when we increase the number of agents to 12, the team improves, as the probabilities of voting for each optimal solution increases (summing up now to 0.982), but they remain evenly distributed. Hence, each option is expected to be seen now 1.6 times.

As we can see, teams of agents with different preferences have a great potential in increasing the number of optimal solutions that we can find, as there is a greater probability of finding a new solution each time we run the system, even though the agents themselves may have a lower probability of finding optimal solutions than the ones in the uniform team.

However, the performance of the team also depends on the team size, and it will not necessarily increase as the team grows (even for the diverse team). For example, in Table 5.3 we can see the pdfs of the diverse team for other team sizes. As we can see, even though the team size increases, for several optimal solutions the probability of voting for them *decreases*. For example, the solution $\langle a_0, a_3 \rangle$ goes from 0.1590 with 6 agents all the way down to 0.0787 with 9 agents. Hence, we will see a lower number of optimal solutions after 10 voting iterations. However, when the number of agents reaches 12, we again have the situation where the total probability is equally divided across all possible optimal solutions.

Hence, the diverse team will, in general, improve as the team size grows, but not for all team sizes. Therefore, we have to increase the team size *carefully* to guarantee optimality. In the next section I formalize my theory.

Solution	$\langle a_0, a_1 \rangle$	$\langle a_0, a_2 \rangle$	$\langle a_0, a_3 \rangle$	$\langle a_1, a_1 \rangle$	$\langle a_1, a_2 \rangle$	$\langle a_1, a_3 \rangle$
6 Agents	0.1590	0.1590	0.1590	0.1590	0.1590	0.1590
7 Agents	0.1107	0.1515	0.1515	0.1466	0.2007	0.2007
8 Agents	0.1118	0.1468	0.1118	0.1806	0.2371	0.1806
9 Agents	0.1046	0.1333	0.0787	0.2171	0.2766	0.1632
10 Agents	0.1065	0.1713	0.1065	0.1641	0.2638	0.1641
11 Agents	0.1339	0.1687	0.1339	0.1666	0.2099	0.1666
12 Agents	0.1638	0.1638	0.1638	0.1638	0.1638	0.1638

Table 5.3: Probability of outputting each possible optimal solution, for different sizes of the diverse team.

5.3.2 Theory

I consider here a team of agents that vote together at each decision point of the design of a product. For the sake of clarity and precision, I present in this section an idealized model. In Section 5.3.3 I generalize my model to more complex situations, and in Section 5.4 I generalize further by performing synthetic experiments.

Let Φ be a set of agents ϕ , and Ω a set of world states ω . Each ω has an associated set of possible actions \mathbf{A}_ω . For example, each world state may represent a parameter of a parametric design problem, and each action may represent a possible value for such parameter. At each world state, each agent ϕ outputs an action a , an optimal action according to the agent’s imperfect evaluation – which may or may not be a truly optimal action. Hence, there is a probability p_j that the agent outputs a certain action a_j . The teams take the action decided by *plurality voting* (i.e, as mentioned in the previous section, the team takes the decision voted by the largest number of agents – I consider ties are broken uniformly at random).

I assume first that the world states are independent, and by taking an optimal action at all world states we find an optimal solution for the entire problem. That is, I assume first that by taking locally optimal decisions at each design decision point, a globally optimal solution is obtained. I generalize this assumption later, in Proposition 5.3.10 (in Section 5.3.3), where I consider design problems with correlated parameters. I will also further discuss this assumption in Section 5.6.

In this chapter my objective goes beyond finding one optimal solution, I want to maximize the number of optimal solutions that we can find. For greater applicability, I consider here agents that output a single action. Hence, we generate multiple solutions by re-applying the voting procedure across all world states multiple times (which are called *voting iterations* – one *iteration* goes across all world states, forming one solution). Formally, let \mathbf{S} be the set of (unique) optimal solutions that we find by re-applying the

voting procedure through z iterations. Our objective is to maximize $|\mathcal{S}|$. I will show that, under some conditions, we can achieve that when $z \rightarrow \infty$ (I study bounded time in Section 5.4).

I consider that at each world state ω there is a subset $\mathbf{Good}_\omega \subset \mathbf{A}_\omega$ of optimal actions in ω . An optimal solution is going to be composed by assigning any $a \in \mathbf{Good}_\omega$ in world state ω – for all world states. Conversely, I consider the complementary subset $\mathbf{Bad}_\omega \subset \mathbf{A}_\omega$, such that $\mathbf{Good}_\omega \cup \mathbf{Bad}_\omega = \mathbf{A}_\omega$, $\mathbf{Good}_\omega \cap \mathbf{Bad}_\omega = \emptyset$. I drop the subscripts ω when it is clear that I am referring to a certain world state.

One fundamental problem is selecting which agents should form a team. By the classical voting theories, one would expect the best teams to be uniform teams composed of multiple copies of the best agent [Conitzer and Sandholm, 2005, List and Goodin, 2001]. Here I show, however, that for design problems uniform teams need very strong assumptions to be optimal, and in most cases they actually converge to always outputting a single solution – an undesirable outcome. However, diverse teams are optimal as long as the team size grows *carefully*, as I explain later in Theorem 5.3.6.

I call a team *optimal* when: (i) $|\mathcal{S}| \rightarrow \prod_\omega |\mathbf{Good}_\omega|$ as $z \rightarrow \infty$, and (ii) all optimal solutions are chosen by the team with the same probability $1/\prod_\omega |\mathbf{Good}_\omega|$. Otherwise, even though the team still produces all optimal solutions, it would tend to repeat already generated solutions whose probability is higher. Since in practice there are time bounds, such condition is fundamental to have as many optimal solutions as possible in limited time. Also note that condition (ii) subsumes condition (i), but I keep both for clarity.

I first consider agents whose pdfs are independent and identically distributed. Let $p_j^{\mathbf{Good}}$ be the probability of voting for $a_j \in \mathbf{Good}$, and $p_k^{\mathbf{Bad}}$ be the probability of voting for $a_k \in \mathbf{Bad}$. Let $n := |\Phi|$ be the size of the team, and N_l be the number of agents that vote for a_l in a certain voting iteration. If $\forall a_j \in \mathbf{Good}, a_k \in \mathbf{Bad}, p_j^{\mathbf{Good}} > p_k^{\mathbf{Bad}}$, the team is going to find one optimal solution with probability 1 as $n \rightarrow \infty$, as I show in the following observation:

Observation 5.3.1 *The probability of a team outputting one optimal solution goes to 1 as $n \rightarrow \infty$, if $p_j^{\mathbf{Good}} > p_k^{\mathbf{Bad}}, \forall a_j \in \mathbf{Good}, a_k \in \mathbf{Bad}$.*

Note that as the agents are independent and identically distributed, we can model the process of pooling the opinions of n agents as a multinomial distribution with n trials (and the probability of any class k of the multinomial corresponds to the probability p_k of voting for an action a_k).

Hence, for each action a_l , the expected number of votes is given by $E[N_l] = n \times p_l$. Therefore, by the *law of large numbers*, if $p_j^{\mathbf{Good}} > p_k^{\mathbf{Bad}} \forall a_j \in \mathbf{Good}, a_k \in \mathbf{Bad}$, we have

that $N_j > N_k$. Hence, the team will pick an action $a_j \in \mathbf{Good}$, in all world states, if n is large enough (i.e., $n \rightarrow \infty$).

However, with a team made of copies of the same agent, the system is likely to lose the ability to generate new solutions as n increases. If, for each ω , we have an action a_m^ω such that $p_m^{Good} > p_j^{Good} \forall a_m^\omega \neq a_j^\omega$, the team converges to picking only action a_m^ω . Hence, $|\mathbf{S}| = 1$, which is a very negative result for design problems. Therefore, contrary to traditional social choice, here it is not the case that increasing the team size always improves performance.

I formalize this notion in Proposition 5.3.2 below, where I also show the conditions for a uniform team to be optimal. Let $p_{Good} := \sum_j p_j^{Good}$ be the probability of picking any action in \mathbf{Good} . I re-write the probability of an action a_j^{Good} as: $p_j^{Good} := \frac{p_{Good}}{|\mathbf{Good}|} + \lambda_j$, where $\sum_j \lambda_j := 0$. Hence, some λ_j are positive, and some are negative (unless they are all equal to 0). Let $\boldsymbol{\lambda}^+$ be the set of $\lambda_j > 0$. Let λ^{High} be the maximum possible value for $\lambda_j \in \boldsymbol{\lambda}^+$, such that the relation $p_j^{Good} > p_k^{Bad}, \forall a_j \in \mathbf{Good}, a_k \in \mathbf{Bad}$ is preserved. I show that when $z \rightarrow \infty$, $|\mathbf{S}|$ is the highest as $\max \boldsymbol{\lambda}^+ \rightarrow 0$, and the lowest (i.e., one) as $\min \boldsymbol{\lambda}^+ \rightarrow \lambda^{High}$. Note that $\max \boldsymbol{\lambda}^+ \rightarrow 0$ represents the situation where the probability is equally divided among all optimal actions, and $\min \boldsymbol{\lambda}^+ \rightarrow \lambda^{High}$ represents the case where one optimal action receives a high probability in comparison with the other optimal actions.

Proposition 5.3.2 *The maximum value for $|\mathbf{S}|$ is $\prod_\omega |\mathbf{Good}_\omega|$. When $z, n \rightarrow \infty$, as $\max \boldsymbol{\lambda}^+ \rightarrow 0$, $|\mathbf{S}| \rightarrow \prod_\omega |\mathbf{Good}_\omega|$. Conversely, as $\min \boldsymbol{\lambda}^+ \rightarrow \lambda^{High}$, $|\mathbf{S}| \rightarrow 1$.*

Proof: As $\max \boldsymbol{\lambda}^+ \rightarrow 0$, $\lambda_j \rightarrow 0, \forall a_j$. Hence, $E[N_j] \rightarrow n \times \frac{p_{Good}}{|\mathbf{Good}|}, \forall a_j \in \mathbf{Good}$. Because ties are broken randomly, at each world state ω , each $a_j \in \mathbf{Good}_\omega$ is selected by the team with equal probability $\frac{1}{|\mathbf{Good}_\omega|}$. As $E[N_j] = E[N_k] \forall a_j, a_k \in \mathbf{Good}$, we have that at each ω it is possible to choose $|\mathbf{Good}_\omega|$ different actions. Hence, there are $\prod_\omega |\mathbf{Good}_\omega|$ possible combinations of solutions. At each voting iteration, ties are broken at each ω randomly, and one possible combination is generated. As $z \rightarrow \infty$, eventually we cover all possible combinations, and $|\mathbf{S}| \rightarrow \prod_\omega |\mathbf{Good}_\omega|$.

Conversely, as $\min \boldsymbol{\lambda}^+ \rightarrow \lambda^{High}$, $E[N_j] \rightarrow n \times p_j^{Good}$ for one fixed a_j such that $p_j^{Good} > p_k^{Good}, \forall a_j \neq a_k \in \mathbf{Good}$. Consequently, $E[N_j] > E[N_k]$, at each ω . Hence, there is no tie in any world state, and the team picks a fixed a_j^ω at each world state. Therefore, even if $z \rightarrow \infty$, $|\mathbf{S}| \rightarrow 1$. Note that I do not say here that the same action is picked across world states (as a_j^ω may differ for each ω), but that the same optimal solution is picked for all voting iterations. ■

Therefore, uniform teams need a very strong assumption to satisfy condition (i): the probability of voting for optimal actions must be uniformly distributed over all optimal

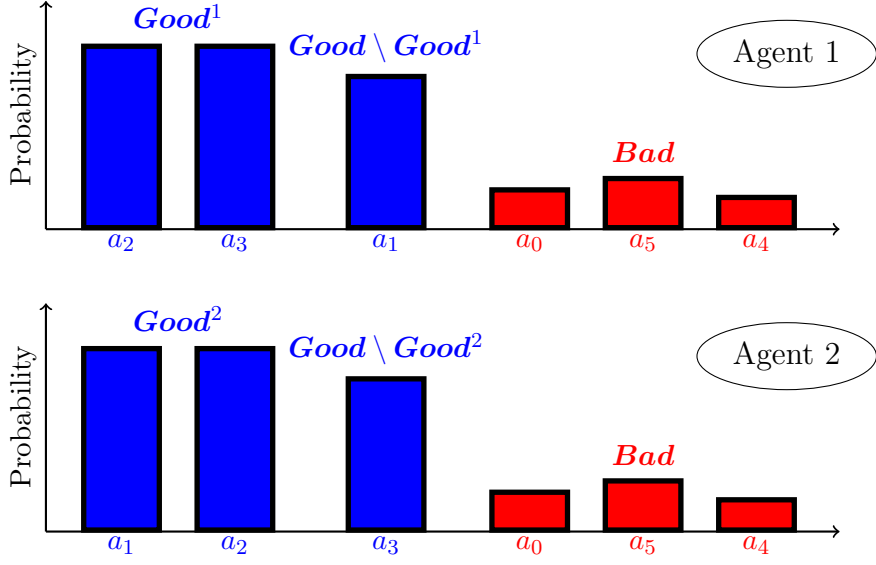


Figure 5.2: Illustrative example of the probability distribution functions of two diverse agents.

actions (i.e., $\max \lambda^+ \rightarrow 0$). If $\max \lambda^+ \rightarrow 0$, condition (ii) is also satisfied as n grows, because of Observation 5.3.1 (i.e., the probability of outputting a suboptimal solution goes to 0) and because of the fact that all actions are equally likely to be chosen; hence each solution is chosen with equal probability $1/\prod_{\omega} |\mathbf{Good}_{\omega}|$.

I show that, alternatively, we can use agents with different “preferences” (i.e., “diverse” agents), to maximize $|\mathcal{S}|$. I consider here agents that have about the same ability in problem-solving, but they prefer different optimal actions. As the agents have similar ability, I consider here the probabilities to be the same across agents, except for the actions in \mathbf{Good} , as each agent ϕ_i has a subset $\mathbf{Good}^i \subset \mathbf{Good}$ consisting of its preferred actions (which are more likely to be chosen than other actions). I denote by p_{ij} the probability of agent ϕ_i voting for action a_j . Hence, I define the pdf of the diverse agents as: $\forall a_j \in \mathbf{Good}^i$, let $p_{\mathbf{Good}^i} := \sum_j p_{ij}$, $p_{ij} := \frac{p_{\mathbf{Good}^i}}{|\mathbf{Good}^i|}$; $\forall a_j \in \mathbf{Good} \setminus \mathbf{Good}^i$, $p_{ij} := \frac{p_{\mathbf{Good}} - p_{\mathbf{Good}^i}}{|\mathbf{Good} \setminus \mathbf{Good}^i|}$; and $\forall a_k \notin \mathbf{Good}^i, a_j \in \mathbf{Good}^i, p_{ij} > p_{ik}$. $\mathbf{Good}^i \cap \mathbf{Good}^l$ (of agents ϕ_i and ϕ_l) is not necessarily \emptyset . The pdfs are strictly defined in this section for the sake of clarity and precision, but in the next section and in my synthetic experiments I generalize further. In Figure 5.2 I show an illustrative example of the pdf of two agents.

Let’s consider we can draw diverse agents from a distribution \mathcal{F} . Each agent ϕ_i has $r < |\mathbf{Good}|$ actions in its \mathbf{Good}^i , and I assume that all actions in \mathbf{Good} are equally likely to be selected to form \mathbf{Good}^i (since they are all equally optimal). Note that r is the same for all agents (as, again, I assume they have the same pdfs, but different preferences),

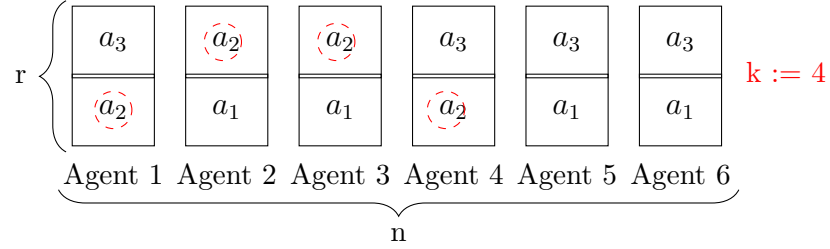


Figure 5.3: Illustrative example of Equation 5.1. Here I show 6 agents ($n := 6$) with 2 preferred actions each ($r := 2$). Each action is in the list of preferences of 4 agents ($k := 4$). As an example, I mark with a dashed circle one of the actions, a_2 .

and that I also cover the case where each agent prefers a single action (which would be $r := 1$).

I will show that by drawing n agents from \mathcal{F} , the team is optimal for large n with probability 1, as long as n is a multiple of a divisor (> 1) of each $|\mathbf{Good}_\omega|$. I will also show that the minimum necessary optimal team size is constant with high probability as the number of world states grow. I start with the following proposition:

Proposition 5.3.3 *If a team of size n is optimal at a world state, then $\gcd(n, |\mathbf{Good}|) > 1$. That is, n and $|\mathbf{Good}|$ are not co-prime.*

Proof: (By contradiction). By the optimality requirement (ii), each action must be in the \mathbf{Good}^i set of the same number of agents. Otherwise, if an action a_i is preferred by a larger number of agents than another action a_j , the team would pick a_i with a larger probability than a_j . Hence, we must have that:

$$n \times r = k \times |\mathbf{Good}|, \quad (5.1)$$

where k is a constant $\in \mathbb{N}_{>0}$. k represents the number of agents that have a given action a_j in its \mathbf{Good}^i . Note that it must be the same for all optimal actions, and therefore we have a single constant.

If n and $|\mathbf{Good}|$ are co-prime, then it must be the case that r is divisible by $|\mathbf{Good}|$. However, this yields $r \geq |\mathbf{Good}|$, which contradicts our assumption. Therefore, n and $|\mathbf{Good}|$ are not co-prime. ■

I illustrate Equation 5.1 with an example in Figure 5.3. In the figure I show 6 agents ($n := 6$), with 2 preferred actions each ($r := 2$), equivalent to the example that I showed before in Table 5.1 (II-b). Note that each action is preferred by 4 agents, and hence I show a case where $k := 4$. As an example, I mark with a dashed circle one of the actions, a_2 . In such case, the team will have an equal probability of picking all optimal actions,

and optimality condition (ii) would be satisfied if the probability of picking suboptimal actions is 0.

If, for example, we now change agent 5 to prefer actions a_2 and a_3 (replacing action a_1 by a_2), then the team would be more likely to pick action a_2 by plurality voting than any other action, and it would be less likely to pick action a_1 than any other action. As the number of voting iterations is limited in actual applications, this situation is not desirable.

Note that we could also have a case where one agent prefers a larger number of actions than others. For example, we could change agent 5 to prefer actions a_1 , a_2 and a_3 . However, as the agent has a limited amount of probability distributed over the actions in the \mathbf{Good}^i set (i.e., $p_{\mathbf{Good}^i}$), we have that necessarily the probability of the agent voting for a_1 and a_3 would drop; hence, the team would pick a_2 more often than the other actions, and a_1 and a_3 less often than the other actions.

This does not mean, however, that there is a single optimal configuration for each number of optimal actions $|\mathbf{Good}|$. There are multiple possible solutions for Equation 5.1, but in any possible solution we will find that the size of the team n and $|\mathbf{Good}|$ are not co-prime.

Proposition 5.3.3 is a necessary but not sufficient condition for optimality. That is, if Equation 5.1 is satisfied, all optimal actions will be selected with the same probability, but it is still necessary for the probability of picking suboptimal actions to go to 0 in order to fully satisfy condition (ii). That will be the case if $p_{\mathbf{Good}^i} = 1$, or if $n \rightarrow \infty$, since $p_j^{\mathbf{Good}} > p_k^{\mathbf{Bad}}$, $\forall a_j \in \mathbf{Good}, a_k \in \mathbf{Bad}$.

Note that Proposition 5.3.3 implies hard restrictions for world states where $|\mathbf{Good}|$ is prime, or for teams with prime size n : if n is prime, $|\mathbf{Good}|$ must be a multiple of n ; and if $|\mathbf{Good}|$ is prime, n must be a multiple of $|\mathbf{Good}|$.

Now let's analyze across a set of world states Ω . For a team of fixed size n , Proposition 5.3.3 applies across all world states. Hence, the team size must be a multiple of a divisor (> 1) of each $|\mathbf{Good}_\omega|$. Note that the pdfs of the agents (and also r) may change according to ω . Let \mathbf{D} be a set containing one divisor of each world state (if two or more world states have a common divisor x , it will be representable by only one $x \in \mathbf{D}$). Hence, $\forall \omega$, $\exists d \in \mathbf{D}$, such that $d \mid |\mathbf{Good}_\omega|$; and $\forall d \in \mathbf{D}$, $\exists \mathbf{Good}_\omega$, such that $d \mid |\mathbf{Good}_\omega|$. There are multiple possible \mathbf{D} sets, from the superset of all possibilities \mathcal{D} .

Therefore, we can now study the minimum size necessary for an optimal team. Applying Proposition 5.3.3 at each world state ω , we have that the minimum size necessary for an optimal team is $n = \min_{\mathbf{D} \in \mathcal{D}} \prod_{d \in \mathbf{D}} d$. Hence, our worst case is when each $|\mathbf{Good}_\omega|$ is a unique prime, as the team will have to be a product of all (unique) optimal action space sizes. This means that:

Proposition 5.3.4 *In the worst case, the minimum team size is exponential in the size of the world states $|\Omega|$. In the best case, the minimum necessary team size is a constant with $|\Omega|$.*

Proof: In the worst case, each added world state ω has a unique prime optimal action space size. Hence, the minimum team size is at least the product of the first $|\Omega|$ primes, which, by the *prime number theorem*, has growth rate $\exp((1 + o(1))|\Omega| \log |\Omega|)$. In the best case, each added \mathbf{Good}_ω has a common divisor with previous ones, and the minimum necessary team size does not change. \blacksquare

However, I show that the worst case happens with low probability, and the best case with high probability. Let G be the maximum possible $|\mathbf{Good}|$, and $M := |\Omega|$. Assume that each world state ω_j will have a uniformly randomly drawn number of optimal actions, denoted as m_j , for all $j = 1, \dots, M$ (i.e., $\forall \omega \in \Omega$). I assume that G is large enough, so that the probability that a given m_j has factor p is $1/p$.

Proposition 5.3.5 *The probability that the minimum necessary team size grows exponentially tends to 0, and the probability that it is constant tends to 1, as $M \rightarrow \infty$.*

Proof: It is sufficient to show that the probability that m_1, \dots, m_{M-1} are all co-prime with m_M tends to 0 as $M \rightarrow \infty$. That is, I show that when adding a new world state ω_M , its $|\mathbf{Good}_\omega|$ will have a common factor with the size of the \mathbf{Good} set of some of the other world states with high probability.

Given any prime p , the probability that at least one of any independently randomly generated $M - 1$ numbers m_1, \dots, m_{M-1} has factor p is $1 - (1 - \frac{1}{p})^{M-1}$, while the probability that one independently randomly generated number m_M has factor p is $\frac{1}{p}$ (for large enough G). Therefore, the probability m_M shares common factor p with at least one of m_1, \dots, m_{M-1} is $\frac{1 - (1 - \frac{1}{p})^{M-1}}{p}$.

The probability that m_M is co-prime with all m_1, \dots, m_{M-1} is:

$$\prod_{\text{all primes } p} \left[1 - \frac{1 - (1 - \frac{1}{p})^{M-1}}{p} \right],$$

which, as $M \rightarrow \infty$, tends to:

$$\prod_{\text{all primes } p} \left(1 - \frac{1}{p} \right) = \frac{1}{\zeta(1)} = 0,$$

where $\zeta(s)$ is the Riemann zeta function. The last equality holds true since:

$$\zeta(1) = \prod_{\text{all primes } p} \frac{1}{1 - p^{-1}} = \sum_{i=1}^{\infty} \frac{1}{i} \rightarrow \infty$$

(as shown by Euler). Hence, with high probability, when adding a new world state ω , $|\mathbf{Good}_\omega|$ will share a common factor with a world state already in Ω . ■

Finally, in the next theorem I show that a diverse team of agents is always optimal as the team grows, as long as it grows *carefully*. That is, I show that for large diverse teams we will be able to satisfy the optimality conditions (i) and (ii), as long as the team size is a multiple of a divisor of $|\mathbf{Good}_\omega|$, $\forall \omega \in \Omega$. Again, I assume that G is large enough, so that the probability that a given m_j has factor p is $1/p$.

Theorem 5.3.6 *Let $\mathbf{D} \in \mathcal{D}$ be a set containing one factor from each \mathbf{Good}_ω . For arbitrary n , the probability that we generate (by drawing from a distribution \mathcal{F}) an optimal team of size n converges to 0 as $|\Omega| \rightarrow \infty$. However, if $n = c \prod_{d \in \mathbf{D}} d$, then the probability that the team is optimal tends to 1 as $c \rightarrow \infty$.*

Proof: For an arbitrary team size n , let \mathbf{P} be the set of its prime factors. Given one $p \in \mathbf{P}$, the probability that p is not a factor of $|\mathbf{Good}_\omega|$ is $1 - 1/p$. The probability that all $p \in \mathbf{P}$ are not factors is: $\prod_p (1 - 1/p)$. As $0 < \prod_p (1 - 1/p) < 1$, the probability that at least one $p \in \mathbf{P}$ is a factor of $|\mathbf{Good}_\omega|$ is $1 - \prod_p (1 - 1/p) < 1$. For $|\Omega|$ tests, the probability that at least one p is a factor in all of them is:

$$\left(1 - \prod_p (1 - 1/p)\right)^{|\Omega|},$$

which tends to 0, as $|\Omega| \rightarrow \infty$. Hence, the probability that $\gcd(n, |\mathbf{Good}_\omega|) = 1$ for at least one ω tends to 1, and the probability that the team can be optimal tends to 0.

However, if:

$$n = c \prod_{d \in \mathbf{D}} d,$$

then $\gcd(n, |\mathbf{Good}_\omega|) \neq 1 \forall \omega \in \Omega$, satisfying the necessary condition in Proposition 5.3.3 at all world states.

Let n_j be the number of agents ϕ_i that have a_j in its \mathbf{Good}^i , and $P(n_j = n_k)$ be the probability that $n_j = n_k$ (that is, the probability that the same number of agents have a_j and a_k in their \mathbf{Good}^i). As each a_j has equal probability of being in a \mathbf{Good}^i , for a large number of drawings from \mathcal{F} (i.e., $c \rightarrow \infty$), we have that $P(n_j = n_k) \rightarrow 1, \forall a_j, a_k \in \mathbf{Good}_\omega, \forall \omega$, by the *law of large numbers*. Hence, each optimal solution will be selected with the same probability.

Moreover, as $p_j^{\mathbf{Good}} > p_k^{\mathbf{Bad}}, \forall a_j \in \mathbf{Good}, a_k \in \mathbf{Bad}$, the probability of picking a suboptimal solution converges to 0 (as $n \rightarrow \infty$ with $c \rightarrow \infty$), and hence the probability of

picking each of the optimal solutions converges to $1/\prod_{\omega} |\mathbf{Good}_{\omega}|$ (satisfying optimality condition (ii)). ■

If it is expensive to test values for n such that Theorem 5.3.6 is satisfied, we can choose $n = c \prod_{\omega} |\mathbf{Good}_{\omega}|$, as it immediately implies the conditions of the theorem. Moreover, if we know the size of all $|\mathbf{Good}_{\omega}|$, we can check if n and $|\mathbf{Good}_{\omega}|$ are co-prime in $O(h)$ time (where h is the number of digits in the smaller number), using the *Euclidean algorithm*. Hence, we can test all world states in $O(|\Omega|h)$ time.

I presented in this section an idealized version of my theory. In practice, the agents' pdfs may differ further than what was considered, and the world states (i.e., parameters of a design problem) may not be completely independent. Hence, in the next section I generalize the theory to more complex situations.

5.3.3 Generalizations

In this section I present several generalizations from my initial idealized model, in order to cover more realistic situations. I start by generalizing my theory to cases where the agents do not have only a probability of $p_{ij} := \frac{p_{\mathbf{Good}^i}}{|\mathbf{Good}^i|}$ or $p_{ij} := \frac{p_{\mathbf{Good}} - p_{\mathbf{Good}^i}}{|\mathbf{Good} \setminus \mathbf{Good}^i|}$ to vote for actions in \mathbf{Good} (depending if the action is in \mathbf{Good}^i or not), but now can have different probabilities distributed over the actions in \mathbf{Good} . Hence, I now model each agent as having a set of \mathbf{Good}^i sets, each with its own probability distributed over the actions in the set. For this generalization, I still consider that the agents have the same pdf, but different preferences. That is, the agents may have different actions at each \mathbf{Good}^i set, but their size and the number of sets is the same across agents.

Hence, I denote each \mathbf{Good}^i set j as ${}_j\mathbf{Good}^i$. Each also has its own $p_{{}_j\mathbf{Good}^i}$ total probability, that will be equally distributed among all actions in ${}_j\mathbf{Good}^i$, in a similar fashion as before. As mentioned, the content of each ${}_j\mathbf{Good}^i$ set may differ across agents, but I consider the $p_{{}_j\mathbf{Good}^i}$ to be the same across agents. Note that the case where each action has a different probability is defined as the situation where each $|{}_j\mathbf{Good}^i| := 1$.

Similarly as before, I consider that each agent ϕ_i has ${}_j r < |\mathbf{Good}|$ actions at each ${}_j\mathbf{Good}^i$, and all actions in \mathbf{Good} are equally likely to be selected to form each ${}_j\mathbf{Good}^i$. In Figure 5.4 I show an illustrative example of the pdf of two agents with multiple ${}_j\mathbf{Good}^i$ sets.

Proposition 5.3.7 *Theorem 5.3.6 still applies under the more general model stated above. That is, if $n = c \prod_{d \in D} d$, then the probability that the team is optimal tends to 1 as $c \rightarrow \infty$.*

Proof: Similarly as before, for each ${}_j\mathbf{Good}^i$ we must have that:

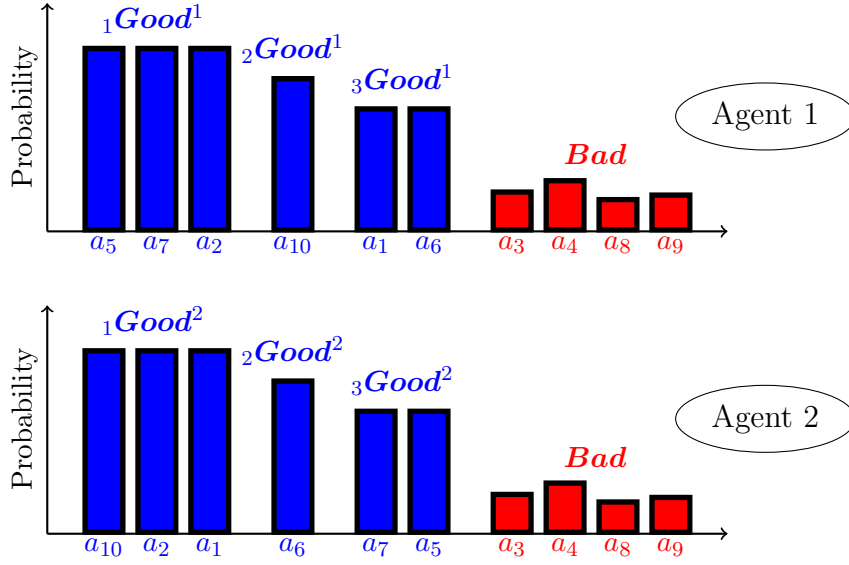


Figure 5.4: Illustrative example of the probability distribution functions of two agents with multiple ${}_j\mathbf{Good}^i$ sets.

$$n \times {}_j r = {}_j k \times |\mathbf{Good}|, \quad (5.2)$$

so that for each ${}_j\mathbf{Good}^i$ we have that ${}_j k$ agents have a given action a in its ${}_j\mathbf{Good}^i$. As the total probability $p_{{}_j\mathbf{Good}^i}$ of each set is the same across agents, we have that each optimal action will be selected by the team with the same probability when Equation 5.2 is satisfied for all ${}_j\mathbf{Good}^i$. Hence, across world states, each optimal solution will also have the same probability of being selected. Similarly as in Proposition 5.3.3, for Equation 5.2 to be satisfied, we must have that n and $|\mathbf{Good}|$ are not co-prime, and that will be true when $n = c \prod_{d \in \mathcal{D}} d$.

Let ${}_j n_l$ be the number of agents ϕ_i that have a_l in its ${}_j\mathbf{Good}^i$, and $P({}_j n_l = {}_j n_m)$ be the probability that ${}_j n_l = {}_j n_m$. Like before, as each a_l has equal probability of being in a ${}_j\mathbf{Good}^i$, for a large number of drawings from \mathcal{F} (i.e., $c \rightarrow \infty$), we have that $P({}_j n_l = {}_j n_m) \rightarrow 1, \forall a_l, a_m \in \mathbf{Good}_\omega, \forall \omega$, by the *law of large numbers*.

Notice that this happens for all ${}_j\mathbf{Good}^i$ sets. Hence, all optimal actions will be selected with the same probability by the team. Like before, as $p_l^{\mathbf{Good}} > p_m^{\mathbf{Bad}}, \forall a_l \in \mathbf{Good}, a_m \in \mathbf{Bad}$, the probability of picking a suboptimal action converges to 0 (as $n \rightarrow \infty$ with $c \rightarrow \infty$), and hence the probability of picking each of the optimal solutions converges to $1 / \prod_\omega |\mathbf{Good}_\omega|$ (satisfying optimality condition (ii)). ■

Now I present my second generalization. I show that Theorem 5.3.6 still applies for agents ϕ_i with different probabilities over optimal actions $p_{\mathbf{Good}^i}$. I consider here a more

general definition of *optimal* team: the difference between the probabilities of picking each optimal solution and $1/\prod_{\omega} |\mathbf{Good}_{\omega}|$ must be as small as possible.

Hence, let p_j^{Φ} be the probability of team Φ picking optimal action a_j , the optimal team is such that $\vartheta := \sum_{a_j} |p_j^{\Phi} - 1/|\mathbf{Good}_{\omega}||$, $\forall a_j \in \mathbf{Good}_{\omega}$ is minimized ($\forall \omega \in \Omega$). I focus here in a single world state ω , as by minimizing ϑ in each world state we are also making the difference between the probability of picking each optimal solution and $1/\prod_{\omega} |\mathbf{Good}_{\omega}|$ as small as possible. Hence, the original definition in the previous section is the case where $\vartheta := 0$.

Proposition 5.3.8 *Theorem 5.3.6 still applies when $|p_{Good^i} - p_{Good^j}| \leq \epsilon$, $\forall \phi_i, \phi_j$, for small enough $\epsilon > 0$.*

Proof: Let Φ be an optimal team, where p_{Good^i} is the same for all agents ϕ_i . Hence, the probability of all actions in \mathbf{Good} being selected by the team is the same. I.e., $p_k^{\Phi} = p_l^{\Phi}, \forall a_k, a_l \in \mathbf{Good}$, and $\vartheta := 0$. Let $\Delta := \sum_{a_k \in \mathbf{Good}} \sum_{a_l \in \mathbf{Good}} |p_k^{\Phi} - p_l^{\Phi}|$ be the difference between the probabilities of the team taking each optimal action. In the rest of the proof we will disturb the probabilities p_{Good^i} of sets of agents, which will change Δ . I focus in studying the variation in Δ , as minimizing the variation in Δ also minimizes the variation in ϑ .

I prove by mathematical induction. Assume we change the p_{Good^i} of x agents ϕ_i , and Δ is as small as possible. Now we will change $x + 1$ agents. Let's pick one agent ϕ_i and increase its p_{Good^i} by $\delta \leq \epsilon$. It follows that $p_k^{\Phi} > p_l^{\Phi}, \forall a_k \in \mathbf{Good}^i, a_l \notin \mathbf{Good}^i$, and the new $\Delta' := \sum_{a_k \in \mathbf{Good}} \sum_{a_l \in \mathbf{Good}} |p_k^{\Phi} - p_l^{\Phi}| > \Delta$.

If we add one more agent ϕ_j , such that $\mathbf{Good}^j \cap \mathbf{Good}^i = \emptyset$, the probability of voting for actions $a_m \in \mathbf{Good}^j$ increases. For small enough ϵ , p_{Good^j} will be too large to precisely equalize the probabilities, and it follows that $p_m^{\Phi} > p_k^{\Phi} > p_l^{\Phi}, \forall a_m \in \mathbf{Good}^j, a_k \in \mathbf{Good}^i, a_l \notin \mathbf{Good}^i \cup \mathbf{Good}^j$, and $\Delta'' := \sum_{a_k \in \mathbf{Good}} \sum_{a_l \in \mathbf{Good}} |p_k^{\Phi} - p_l^{\Phi}| > \Delta'$. The same applies for each newly added agent, until we have a new team such that $n = c \prod_{d \in \mathcal{D}} d$ (again, satisfying the conditions of the theorem).

The base case follows trivially. If we did not change the probability of any agent (i.e., $x := 0$), and we now increase p_{Good^i} of a single agent ϕ_i , $p_k^{\Phi} > p_l^{\Phi}, \forall a_k \in \mathbf{Good}^i, a_l \notin \mathbf{Good}^i$, and $\Delta' > \Delta$. By the same argument as before, adding more agents will only increase Δ' , until $n = c \prod_{d \in \mathcal{D}} d$. \blacksquare

Thirdly, I also generalize to the case where the number of preferred actions r changes for each agent. I consider that the number of actions in the \mathbf{Good}^i of each agent ϕ_i (r^i) is decided according to a uniform distribution on the interval $[1, r']$.

Proposition 5.3.9 *If $n = r' \times c \prod_{d \in \mathcal{D}} d$, the probability that the team is optimal $\rightarrow 1$ as $c \rightarrow \infty$.*

Proof: For large n , the number of agents with $r^i = 1, \dots, r'$ is the same. Therefore, if for each subset $\Phi^i \subset \Phi$, such that $r^\phi = i, \forall \phi \in \Phi^i$, we have that $p_k^{\Phi^i} = p_l^{\Phi^i}, \forall a_k, a_l \in \mathbf{Good}$, we will have that $p_k^\Phi = p_l^\Phi, \forall a_k, a_l \in \mathbf{Good}$. Given an optimal team of size n , we have r' subsets Φ^i of size n/r' each. It follows by Theorem 5.3.6 that $n/r' = c \prod_{d \in D} d$, and:

$$n = r' \times n/r' = r' \times c \prod_{d \in D} d,$$

hence n also follows the necessary conditions in Proposition 5.3.3. Similarly as in Theorem 5.3.6, as $n \rightarrow \infty$ with $c \rightarrow \infty$, the probability of picking a suboptimal solution converges to 0, and the probability of picking each of the optimal solutions converges to $1/\prod_\omega |\mathbf{Good}_\omega|$ (satisfying optimality condition (ii)). ■

Lastly, I discuss the assumption that world states are independent. In design problems they could actually be correlated. Hence, I present below a constructive proof showing that we can still use our model to study design problems with correlated parameters.

Proposition 5.3.10 *The previous results still apply for design problems with correlated parameters.*

Proof: Let's consider a design problem with a set Υ of parameters v . We can divide Υ in Υ_k sets, where all $v \in \Upsilon_k$ are correlated, but v_i and v_j are independent, $\forall v_i \in \Upsilon_i, v_j \in \Upsilon_j, i \neq j$. That is, all parameters v in a Υ_k set are correlated, but the parameters between two different Υ_k sets are independent. This can always be performed, as in the worst case where all parameters are correlated, we can have a single $\Upsilon_k := \Upsilon$.

Now, instead of modeling each design parameter v as a world state ω (as in my original model), we can model each set Υ_k as a world state ω . Hence, instead of an action a being one value assigned to a parameter v , an action a now represents one full combination of values to each v_k in a set Υ_k .

Hence, instead of voting at each parameter v , each agent ϕ_i now votes for one combination of value assignments (of correlated parameters) at each set Υ_k . As all sets Υ_k are independent, we still have agents voting for independent world states ω and the previous results still apply. In the worst case, where all parameters of the problem are correlated, we would have agents voting for entire solutions, and the model would be considered as having a single world state ω . ■

In the next section I perform synthetic experiments with agents whose pdfs differ, to further generalize over my theory, and I show that diverse teams still significantly outperform uniform teams.

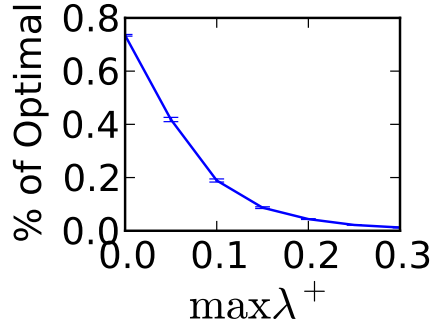


Figure 5.5: Percentage of optimal solutions found by uniform teams as $\max \lambda^+$ grows. Note how $\max \lambda^+ := 0$ gives the best result.

5.4 Synthetic Experiments

I run synthetic experiments, where I simulate design agents and evaluate diverse and uniform teams (henceforth *diverse* and *uniform*). I randomly create pdfs for the agents, and simulate voting iterations across a series of world states. I repeat all my experiments 100 times, and in the graphs I plot the average and the confidence interval of my results (according to a *t-test* with $p := 0.01$). I run 1000 voting iterations (z), and measure how many optimal solutions the team is able to find. I study a scenario where the number of actions ($|\mathbf{A}|$) := 100, and the number of optimal actions per world state ($|\mathbf{Good}_\omega|$) is, respectively: $\langle 2, 3, 5, 5, 5 \rangle$, in a total of 750 optimal solutions.

At each repetition of my experiment, I randomly create a pdf for the agents. I start by studying the impact of $\max \lambda^+$ in *uniform*. When creating the *uniform* team, the total probability of playing any of the optimal actions (i.e., $p_{\mathbf{Good}}$) is randomly assigned (uniform distribution) between 0.6 and 0.8. I fix the size of the team (25) and evaluate different $\max \lambda^+$ in Figure 5.5. As expected from Proposition 5.3.2, for $\max \lambda^+ := 0$ the system finds the highest number of optimal solutions; and as $\max \lambda^+$ increases, it quickly drops.

I then study the impact of increasing the number of agents, for *uniform* and *diverse*. To generate a *diverse* team, I draw randomly a r_ω in an interval U for each world state, that will be the size of $|\mathbf{Good}^i|$. I study three variants: *diverse**, where $U := (0, |\mathbf{Good}_\omega|)$; *diverse*, where $U := (0, |\mathbf{Good}_\omega|)$, and *diverse Δ* , where I allow agents to have different r_ω^i , also drawn from $(0, |\mathbf{Good}_\omega|)$. I independently create pdfs randomly for each agent ϕ_i . For each agent I draw a number between 0.6 and 0.8 to distribute over the set of optimal actions, and randomly decide r_ω actions to compose its \mathbf{Good}^i set. I distribute equally 80% of the probability of voting over optimal actions on the actions of that set.

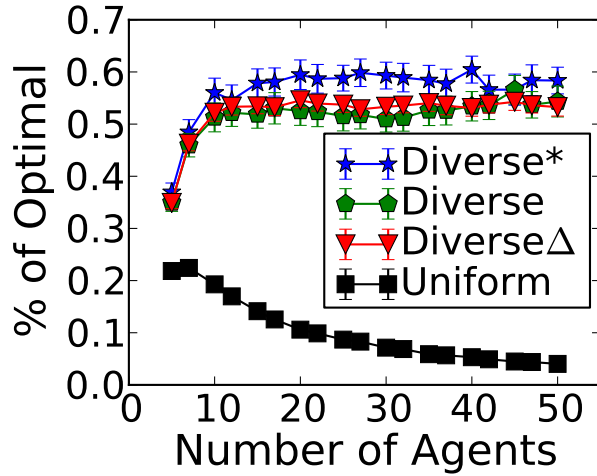


Figure 5.6: Percentage of optimal solutions as the number of agents grows. The uniform teams *decrease* in performance, while multiple variations of diverse teams improve, but with diminishing returns.

As we can see, in Figure 5.6, the number of solutions *decreases* for *uniform* as the number of agents grows. Normally, in social choice, we expect the performance to improve as teams get larger, so this is a novel result. It is, however, expected from our Proposition 5.3.2. *Diverse*, on the other hand, improves in performance for all 3 versions, as predicted by my theory. However, the system seems to converge for a fixed z , as the performance does not increase much after around 20 agents.

Hence, in Figure 5.7 I study larger *diverse* (continuous line) and *diverseΔ* teams (dashed line), going all the way up to 1800 agents. I also study four different number of voting iterations (z , shown in the figure by different lines): 1000, 2000, 3000, 4000. As we can see, although adding more agents was not really improving the performance in the experimental scenario under study, there is clearly a statistically significant improvement ($p < 0.01$) by increasing the number of voting iterations, with the system improving from finding around 53% of the optimal solutions, all the way up to finding more than 80% of them. However, there is a *diminishing returns* effect, as the impact of adding more iterations decreases as the actual number of iterations grow larger. We also note that *diverseΔ* is better than *diverse*, and the difference increases as z grows.

As we can see, although theoretically possible, it is still a challenge to have a system that can find all the possible optimal solutions. Moreover, it would be expensive to pool the votes of agents through a large number of voting iterations. However, as I show next, we can actually approximate this process in a real system, by pooling only a small number of solutions from each agent, and executing many voting iterations by aggregating

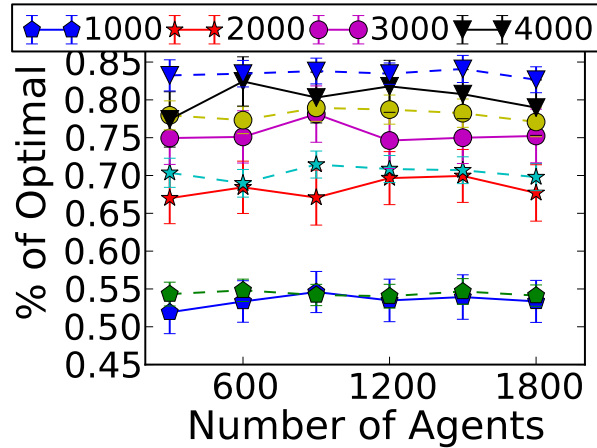


Figure 5.7: Percentage of optimal solutions found by large teams of diverse agents.

different combinations of these solutions. In the next section I show executions in a real design system.

5.5 Experiments in Architectural Design

5.5.1 Architectural Design Domain

I study a real system for architectural building design. This is a fundamental domain, since the design of a building impacts its energy usage during its whole life-span [Bogenstätter, 2000, Lin and Gerber, 2014, Echenagucia et al., 2015]. I use Beagle [Gerber and Lin, 2013], a multi-objective design optimization software that assists users in the early stage design of buildings. Hence, the experiments presented here were run in an actual system that performs expensive energy evaluations over complex architectural designs, and represent months of experimental work.

First, the designer creates a parametric design, containing (as discussed in Section 5.2) a set of parameters that can be modified within a specified range, allowing the creation of many variations. The ranges are defined according to the legislation (i.e., setback, maximum height, etc), or the intention of the designer (for example, the general shape of the building).

I use designs from Gerber and Lin [2013]: *base*, a simple building type with uniform program (i.e., tenant type); *office park*, a multi-tenant grouping of towers; and *contemporary*, a double “twisted” tower that includes multiple occupancy types, relevant to contemporary architectural practices. I show the designs in Figure 5.8.

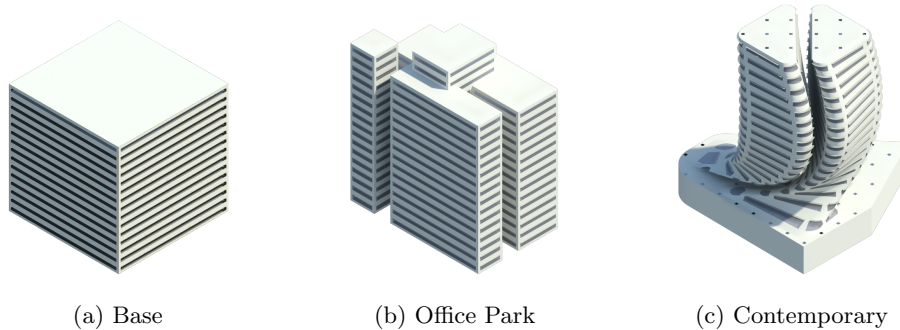


Figure 5.8: Parametric designs with increasing complexity used in our experiments.

Beagle uses a genetic algorithm (GA) to optimize the building design based on three objectives: energy efficiency, financial performance and area requirements. In detail, the objective functions are: $S_{obj} : \max SPCS$; $E_{obj} : \min EUI$; $F_{obj} : \max NPV$. SPCS is the Spatial Programming Compliance Score, EUI is the Energy Use Intensity and NPV is the Net Present Value, defined as follows.

SPCS defines how well a building conforms to the project requirements (by measuring how close the area dedicated to different activities is to a given specification). Let \mathbf{L} be a list of activities (in our designs, $\mathbf{L} = \langle \text{Office, Hotel, Retail, Parking} \rangle$), $area(l)$ be the total area in a building dedicated to activity l and $requirement(l)$ be the area for activity l given in a project specification. SPCS is defined as: $SPCS := 100 \times \left(1 - \frac{\sum_{l \in \mathbf{L}} |area(l) - requirement(l)|}{|\mathbf{L}|} \right)$

EUI regulates the overall energy performance of the building. This is an estimated overall building energy consumption in relation to the overall building floor area. The process to obtain the energy analysis result is automated in Beagle through Autodesk Green Building Studio (GBS) web service.

Finally, **NPV** is a commonly used financial evaluation. It measures the financial performance for the whole building life cycle, given by: $NPV := \left(\sum_{t=1}^T \frac{c_t}{(1+r)^t} \right) - c_0$, where T is the Cash Flow Time Span, r is the Annual Rate of Return, c_0 is the construction cost, and $c_t := \text{Revenue} - \text{Operation Cost}$.

Many options affect the execution of the GA, including: initial population size, size of the population, selection size, crossover ratio, mutation ratio, maximum iteration. Further details about Beagle are at Gerber and Lin [2013].

In the end of the optimization process, the GA outputs a set of solutions. These are considered “optimal”, according to the internal evaluation of the GA, but are not necessarily so. As in my theory, for each parameter the assigned value is going to be one

Agent	PZ	SZ	CR	MR
Agent 1	12	10	0.8	0.1
Agent 2	18	8	0.6	0.2
Agent 3	24	16	0.55	0.15
Agent 4	30	20	0.4	0.25

Table 5.4: GA parameters for the diverse team. Initial Population and Maximum Iteration were kept as constants: 10 and 5, respectively. PZ = Population Size, SZ = Selection Size, CR = Crossover Ratio, MR = Mutation Ratio.

of the optimal ones with a certain probability. In fact, most of the solutions outputted by the GAs are later identified as sub-optimal and eliminated in comparison with better ones found by the teams.

I model each run of the GA as an agent ϕ . Each parameter of the parametric design is a world state ω , where the agents decide among different actions \mathbf{A} (i.e., possible values for the current parameter). My model assumes independent multiple voting iterations across all world states. However, in general it could be expensive to pool agents for votes in a large number of iterations. Therefore, in order to test the applicability of the predictions of my model in more realistic scenarios, in my experiments I actually pool only 3 solutions per agent, but run multiple voting iterations by aggregating over all possible combinations of them. That is, at each combination I pick one solution per agent, and vote across all the design parameters, in a total of 81 voting iterations with 4 agents. Nevertheless, as I show next, the predictions of my model are verified in my empirical experiments, presented next.

5.5.2 Empirical Results

I run experiments across the different parametric designs shown in Figure 5.8. These are designs with increasing complexity. More details about the designs and the meanings of each parameter are available in Gerber and Lin [2013]. I create 4 different agents, using different options for the GA, as shown in Table 5.4.

Contrary to the previous synthetic experiments, we are dealing here with real (and consequently complex) design problems. Hence, the true set of optimal solutions is unknown. I approach the problem in a comparative fashion: when evaluating different systems, I consider the union of the set of solutions of all of them. That is, let \mathbf{H}_x be the set of solutions of system x ; I consider the set $\mathcal{H} := \bigcup_x \mathbf{H}_x$. I compare all solutions in \mathcal{H} , and consider as optimal the best solutions in \mathcal{H} , forming the set of optimal solutions \mathcal{O} . I use the concept of Pareto dominance: the best solutions in \mathcal{H} are the ones that *dominate* all other solutions (i.e., they are better in all 3 objectives). As I know which

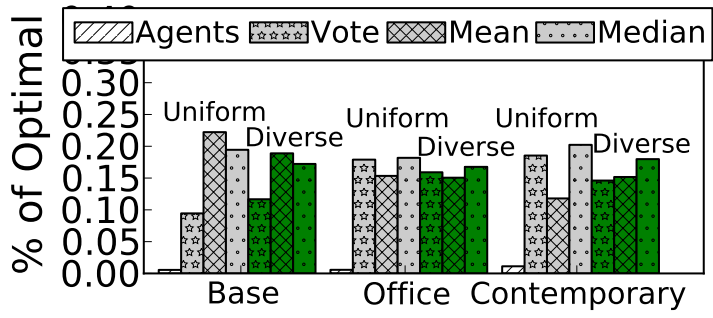
system generated each solution $o \in \mathcal{O}$, I estimate the set of optimal solutions \mathbf{S}_x of each system.

Although my theory focuses on *plurality voting* as the aggregation methodology, in order to have a more thorough experimental study, I also present results using the *mean* and the *median* of the opinions of the agents. That is, given one combination (a set of one solution from each agent), I also generate a new solution by calculating the *mean* or the *median* of the values from each agent across all parameters (i.e., world states). Also, when performing the voting aggregation (*vote*), I consider values that are the same up to 3 decimal places as equal.

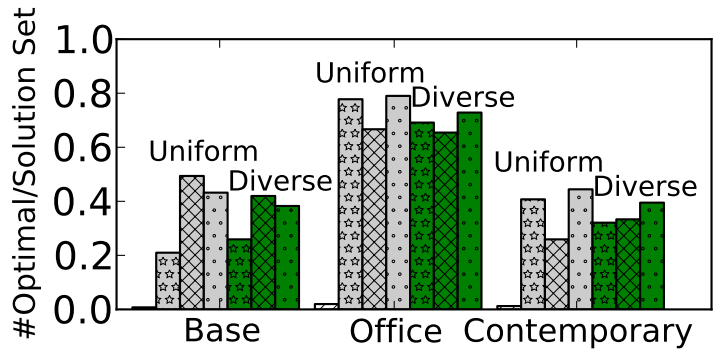
Concerning *uniform*, I evaluate a team composed of copies of the “best” agent. By “best”, I mean the agent that finds the highest number of optimal solutions. According to Proposition 5.3.2, such an agent should be the one with the lowest $\max \lambda^+$, and we can predict that voting among copies of that agent generates a large number of optimal solutions. Hence, for each design, I first compare all solutions of all agents (i.e., construct \mathcal{H} as the union of the solutions of all agents), to estimate which one has the largest set of optimal solutions \mathbf{S} . I, then, run that agent multiple times, creating *uniform*. For *diverse*, I consider one copy of each agent in Table 5.4.

I aggregate the solutions of *diverse* and *uniform*. I run 81 aggregation iterations (across all parameters/world states), by selecting 3 solutions from each agent ϕ_i , in its set of solutions \mathbf{H}_i , and aggregating all possible combinations of these solutions. I evaluate together the solutions of all agents and all teams (i.e., I construct \mathcal{H} with the solutions of all systems), in order to estimate the size of \mathbf{S}_x of each system.

Since the true optimal solutions set is unknown, I first plot the percentage of unique solutions found by each system in relation to the total number of unique optimal solutions in \mathcal{H} . Hence, in Figure 5.9 (a), I show the percentage of optimal solutions for all systems, in relation to $|\mathcal{O}|$. For clarity, I represent the result of the individual agents by the one that had the highest percentage. As we can see, in all parametric designs the teams find a significantly larger percentage of optimal solutions than the individual agents. The agents find less than 1% of the solutions, while the teams are in general always close to or above 15%. In total (considering all aggregation methods and all agents), for all three parametric designs the agents find only about 1% of the optimal solutions, while *uniform* finds around 51% and *diverse* 47%. Looking at *vote*, in *base diverse* finds a larger percentage of optimal solutions than *uniform* (around 9.4% for *uniform*, while 11.6% for *diverse*). In *office park* and *contemporary*, however, *uniform* finds more solutions than *diverse*. Based on Proposition 5.3.2, we expect that this is caused by the best agent having a lower $\max \lambda^+$ in *office park* and *contemporary* than in *base*.



(a) Percentage in relation to all solutions found by all systems.



(b) Percentage in relation to the number of solutions of each system.

Figure 5.9: Percentage of optimal solutions of each system. The teams find a much larger percentage than the individual agents.

Figure 5.9 (b) shows the percentage of optimal solutions found, in relation to the size of the set of evaluated solutions of each system. That is, let \mathbf{O}_x be the set of optimal solutions of system x , in \mathcal{O} . I show $\frac{|\mathbf{O}_x|}{|\mathbf{H}_x|}$. Concerning *vote*, the teams are able to find a new optimal solution around 20% of the time for *base*, around 73% of the time for *office park* and around 36% of the time for *contemporary*. Meanwhile, for the individual agents it is close to 0%. We can see that teams have great potential for generating new optimal solutions, as expected from my theory. However, as studied in my synthetic experiments, we can expect some *diminishing returns* when increasing the number of voting iterations. I show examples of solutions created by the teams in Figure 5.10.

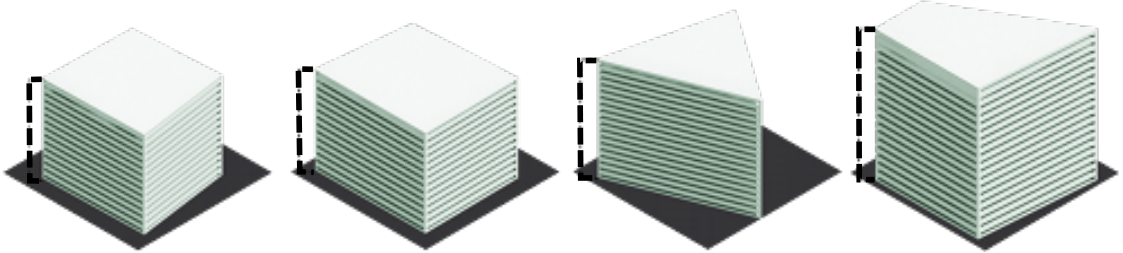
It is interesting to note that the performance of the teams is much higher for *office park* than for the other two parametric designs. In *base* and *contemporary*, the building mass is parametrized into a single volume, while in *office park* the building mass has multiple volumes. Hence, a possible explanation is that the division in multiple volumes facilitated the generation of multiple optimal solutions, since these can be combined in many different ways.

I also plot in Figure 5.11 (a) the percentage of solutions that were reported to be optimal by each agent, but were later discovered to be suboptimal by evaluating \mathcal{H} . A large amount of solutions are eliminated, close to 100%, helping the designer to avoid making a poor decision, and increasing her confidence that the set of optimal solutions found represent well the “true” Pareto frontier. Moreover, I test for duplicated solutions across different aggregation methods, different teams and different agents. The number is small: only 4 in *contemporary*, and none in *base* and *office park*. Hence, we are providing a high coverage of the Pareto frontier for the designer. I show the total number of optimal solutions in Figure 5.11 (b). We reconfirm here that in the *office park* design, where the building mass is divided in multiple volumes, we could generate a larger number of optimal solutions.

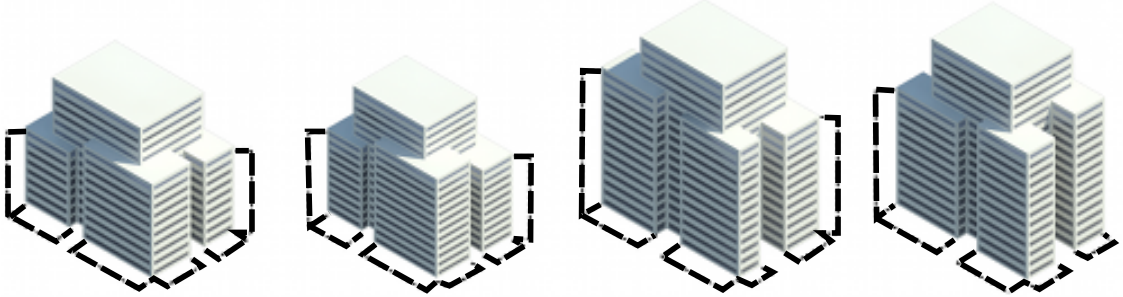
Finally, to better study the solutions proposed by the agents and teams, I plot all the optimal solutions in the objectives space in Figure 5.12, where I show that the solutions give a good coverage of the Pareto frontier.

5.6 Discussion

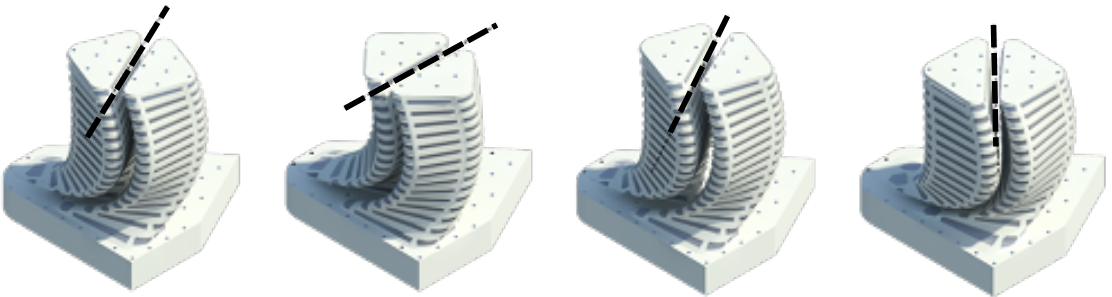
I present in this chapter a new model of teams of voting agents for design problems. I propose as the main objective of such system to present for human evaluation as many optimal solutions as possible. Hence, in this chapter I present a view of the role of artificial intelligence (AI) in the creative process as generating a large number of optimal



(a) Base. Shaded area shows variance of building's footprint in relation to site. Dashed line indicates height variance.



(b) Office Park. Dashed line shows variance in volume.



(c) Contemporary. Line shows variance in orientation.

Figure 5.10: Some building designs generated by the teams.

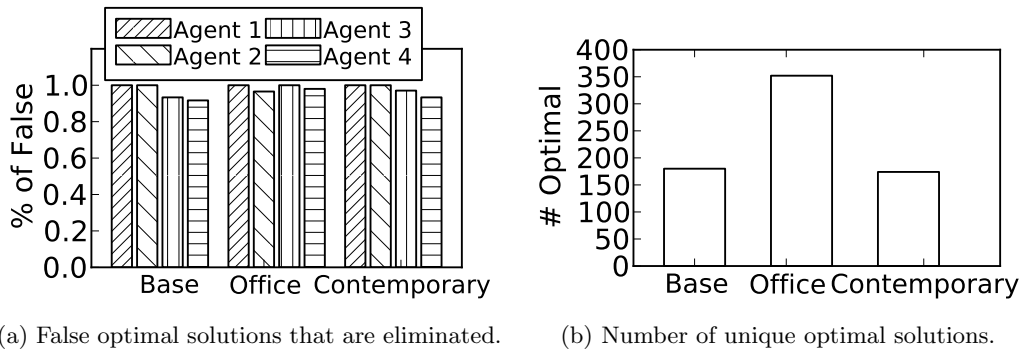


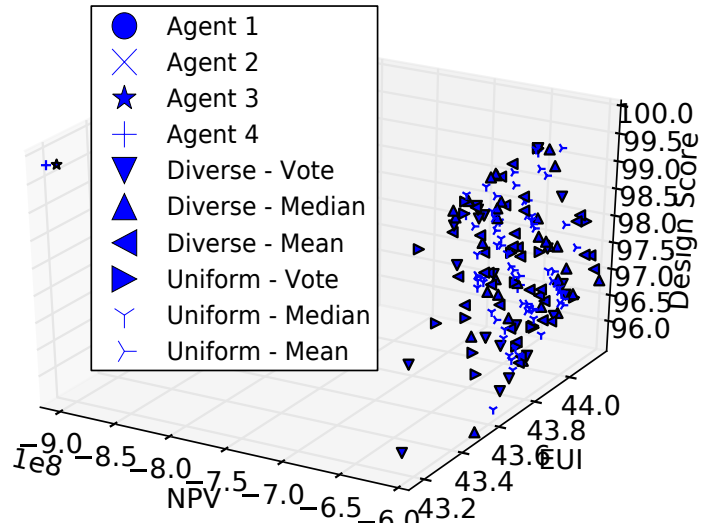
Figure 5.11: Additional analysis. I show here that many falsely reported optimal solutions are eliminated by the team of agents, and also that the teams provide a large number of optimal solutions to the designer.

solutions, so that a human can later take an aesthetic choice (also based on a quantitative evaluation of different objectives, solving complex trade-offs).

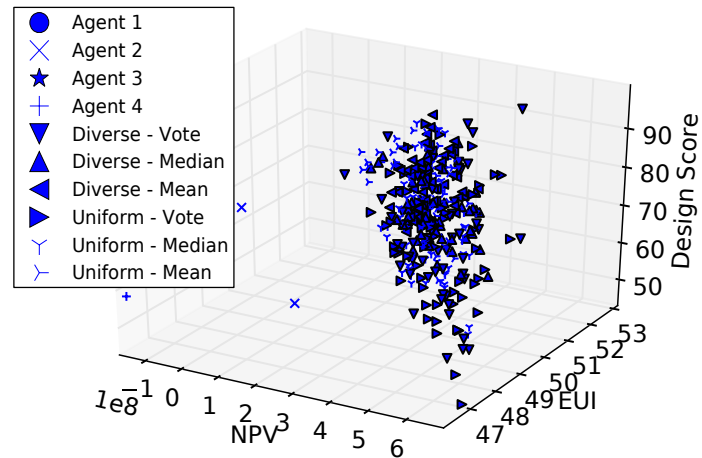
A human designer also needs to input an initial parametric design (such as the ones in Figure 5.8) for the agents to vote for different combinations of values for such design. Hence, this chapter views the role of AI in the creative process as enabling a collaborative work between a human and algorithms. I study here the potential of voting systems to be *creative* by generating a large number of optimal solutions, but the entire creative process is actually a collaboration between the voting system and the designer, and the aesthetic evaluation of quality/beauty is assigned by a human. Hence, the AI system is still in the role of optimizing quantifiable metrics (cost, energy efficiency, etc), that would be too burdensome for a human to optimize, while the human designer assumes the aesthetic evaluation work (which I argue that cannot be properly taken by the machine), besides solving trade-offs between objectives that may not be well formalized, and hence cannot be handled by the computer.

The role of AI in the creative process is an important topic of current discussion. For example, d’Inverno and McCormack [2015], in a very recent position paper, divide the AI research in two different types: what they ironically call “Heroic AI”, when researchers try to build systems that are in the role of the whole creative process; and “Collaborative AI”, when a computational system works in collaboration with a human in the creative process. This chapter, therefore, is aligned with the second classification of d’Inverno and McCormack.

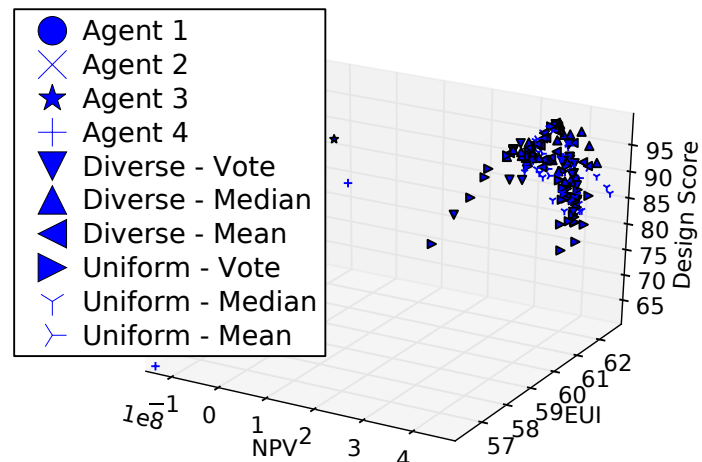
Another important topic of discussion is how to evaluate AI systems that handle creative processes. There is not yet a widely accepted evaluation metric in the literature. Some researchers use tests inspired by the “Turing test”, and evaluate if humans can distinguish if a creative output (such as music), was produced by a human or by a



(a) Base



(b) Office Park



(c) Contemporary

Figure 5.12: All the optimal solutions in the objectives space.

computer [Burnett et al., 2012]. Other works evaluate in qualitative terms, by reporting the impressions of a human when using the system to create [Berman and James, 2015]. Others, such as Machado et al. [2015], uses evolutionary algorithms to generate creative objects (in their case, ambiguous images), and uses the fitness function of the evolutionary algorithm as a quantitative evaluation metric. However, in the end they still need to manually look over the “optimal” solutions found by the algorithm and make a qualitative evaluation if the system was really able to create what it was supposed to. Hence, the quantitative evaluation still fails to really represent a measurable quality of the system.

In this chapter I propose the number of optimal alternatives as a quantitative evaluation metric for a “Collaborative AI” system. Of course that may not be the best quantitative metric for all creative processes where AI is involved, but it is one possible evaluation metric that could be used for systems where the aesthetic choice is left for the human.

It could still be an issue that we may not want to overload a human with too many solutions. Although mathematically all the solutions are of equal “optimal” quality, the human may prefer one objective over another (if in a multi-objective optimization problem), or may want to only evaluate highly different solutions in the aesthetic sense. This situation is, however, much better than if the human had to pick one solution in the space of all possible solutions (optimal and suboptimal). The proposed system already helps to reduce the space to only the solutions that matter, by presenting only optimal solutions for human evaluation.

I propose, hence, that such voting systems that maximize the number of optimal solutions could be used as a building block of a bigger “Collaborative AI” system. Its output could be the input of another system that cluster and organize the proposed solutions, in order to present them to a human designer in a manageable way. As mentioned previously, systems that organize a large number of solutions for human evaluation is currently an active research topic in computational design [Erhan et al., 2014, Smith et al., 2010].

Additionally, my model assumes independent world states, which are initially modeled as the parameters of a parametric design problem. Some design problems, however, may have correlated parameters. I discuss an extension of my model in Proposition 5.3.10, where the parameters are grouped into independent groups. In the worst case of such extension, if all parameters are correlated, the agents would vote for entire solutions, rather than each parameter.

Besides, a theoretical model allows us to make predictions about a real system, but most often is only an approximation of reality (as actual systems are normally overly complex to be modeled and analyzed in detail). Hence, even though the parameters of a problem may not be completely independent, they (or some subsets of them, as in

my extension) may still have some degree of independence, so that my model will still approximate the real execution.

In light of Proposition 5.3.10, one may also question the need for world states in my model, instead of simply always voting for entire solutions, which would work for any design problem. I argue, however, that we should use multiple world states whenever possible/reasonable, as it has the greatest potential in creating new solutions. With a single world state, we will always pick one solution proposed by one of the agents, while with multiple states we can generate solutions that were not individually proposed by none of the agents (since at each world state we may select actions proposed by different subsets of agents).

There are, of course, even more complex problems whose solutions cannot even be properly evaluated by any metric, such as the “wicked problems” coined by Rittel and Webber [1973]. Such problems seem to be beyond the scope of my theory, as we need to be able to evaluate a set of solutions with some well defined metric, so that a set of “optimal” solutions can be found. At the very least, we would need an approximate measurable quality for each solution, which could be obtained, for example, with computer simulations.

Finally, this chapter discusses the importance of having diverse agents, but not how such set of agents with different preferences can be effectively obtained. In Section 5.5 I create different agents by using different settings in a GA software (Beagle), but that may not yet be the best way to obtain such diverse teams.

Creating and evaluating different diversification strategies is still avenue for future work. Perhaps the evaluation function of each GA could be tweaked to bias its search to different portions of the search space. Some of my preliminary results [Marcolino et al., 2014a], however, show that this should be done with care, as giving one unique objective for each agent to optimize was not as effective as using different parametrizations for each GA in the way that I present in this chapter.

Essentially, the best diversification strategy may depend on each specific design problem. For some problems it could be the case that a bias towards specific kinds of solutions could be programmed for each agent, or different algorithmic strategies may converge to different solutions. A deeper study on how diverse agents can be effectively generated for different design problems is, however, beyond the scope of this thesis. Nevertheless, this chapter sets formal and theoretical guidelines for the community to better develop, in future work, strong diverse voting teams for design.

5.7 Conclusion

Design imposes a novel problem to social choice: maximize the number of optimal solutions. This problem is of fundamental importance, since designers need a large space of alternatives in order to take a choice. Ideally every solution a designer examines should be optimal according to measurable metrics, so that she can then solve complex trade-offs that may not be formalized and/or decide according to aesthetics.

Hence, I present a new model for agent teams, that shows the potential of a system of voting agents to be *creative*, by generating a large number of optimal solutions to the designer. My analysis, which builds a new connection with *number theory*, presents several novel results: (i) uniform teams are in general suboptimal, and converge to a unique solution; (ii) diverse teams are optimal as long as the team size is increased *carefully*; (iii) the minimum optimal team size is constant with high probability; (iv) the worst case for teams is a prime number of optimal actions.

I also extend my theory to more general situations, covering cases where agents have greater variability on their probabilities of voting for each optimal action, considering not only a variability across different actions but also across different agents. Most importantly, I also show one way to generalize my theory to cover design problems with correlated parameters, by considering independent parameter sets.

I further study my model with synthetic experiments, which evaluate teams of agents with bounded time and relaxed assumptions, as I allow the probability distribution functions of the agents to vary. The experiments explore three variations of diverse teams and show that they all perform well, increasing in performance as the number of agents and/or the number of voting iterations grow, although with *diminishing returns*. Uniform teams, on the other hand, *decreased* in performance as the number of agents increased, as expected by the theory.

I present results in conceptual architectural design, where I study a real team of *genetic algorithm* agents, which propose different alternatives to the design of buildings, optimizing for energy-efficiency, cost and project requirements. I show that teams find a large number of solutions for designing energy-efficient buildings across three different parametric designs of increasing complexity. Moreover, I find that when voting a diverse team is able to find a larger set of solutions than a uniform team in the real system for one of the design problems, a result that could be explained by my theory. I also noticed that we could find a larger number of optimal solutions in the parametric design where the building mass was divided in multiple volumes, which could indicate that these kind of designs are preferable when using voting systems.

Finally, I present an extensive discussion about how this work fits in the context of the study of creativity within AI, and I argue that my model can be considered as “Collaborative AI”, where humans and algorithms work together in the process of creation. I also propose that voting systems which maximize the number of optimal solutions could be combined with another system that cluster and filter the solutions, and presents them in a manageable way to a designer. Such systems are a current topic of research in the computational design literature.

This chapter, hence, lays a first theoretical foundation for the study of multi-agent voting teams for design problems. Further extensions of this model, and a deeper study on how to methodologically create diverse agents for design are exciting venues for the multi-agents and/or computational design community perform future work.

Part III

Aggregation of Opinions

Chapter 6

RANKED VOTING

*A difference of opinions just means you and I
On different subjects do not see eye to eye
And who is wrong or right is not for me to say
Since we do look at life in a different way.
(Francis Duggan)*

6.1 Introduction

Ranked voting is an active topic of research in the social choice literature [Caragiannis et al., 2013, Xia and Conitzer, 2011a,b, Soufiani et al., 2012, Baharad et al., 2011, Mao et al., 2013]. However, there are not many systems actually using ranked voting approaches, and plurality is still the most common aggregation methodology. Additionally, it is not yet clear how to extract rankings from existing agents, and how to methodologically generate diverse teams of agents in order to aggregate their opinions.

In this chapter, I study the performance of ranked voting approaches in two real systems: Computer Go and Influence Maximization in social networks using PSINET agents. I analyze experimentally the performance of classical ranked voting rules, and discuss different ways to extract rankings from existing agents. Additionally, I will study the performance of large diverse teams of agents, by generating random parametrizations of one base agent.

This chapter encompasses two of my publications: Jiang et al. [2014] and Yadav et al. [2015]. Additional experimental and theoretical results can be found in those publications, but in this thesis I will focus on my main contributions, and hence only some of the results of these publications is presented and discussed here. Moreover, contrary to the rest of the thesis, this chapter does not introduce new models, but rather presents techniques for

team generation and ranking extraction, and experimental results in real systems. The main focus of the experiments will be in the Computer Go domain, but I will also present results in social networks, when handling the influence maximization problem.

This chapter is organized in the following way: I briefly introduce the social networks domain that I consider in this chapter below (Section 6.1.1). Then, Section 6.2 and Section 6.3 will introduce the team generation and ranking extraction methodologies, in the context of Computer Go. Section 6.4 will discuss the social network domain in more detail, including the generation of voting agents and the aggregation methodologies in that context. Section 6.5 presents my experimental results: First, I will show results in the Computer Go domain in Sections 6.5.1 and 6.5.2; in Section 6.5.3 I present the results in the social networks domain.

6.1.1 Social Networks

Homelessness affects ~ 2 million youths in USA annually, 11% of whom are HIV positive, which is 10 times the rate of infection in the general population [Aidala and Sumartojo, 2007]. Peer-led HIV prevention programs such as POL [Kelly et al., 1997] try to spread HIV prevention information through network ties and recommend selecting intervention participants based on Degree Centrality (i.e., highest degree nodes first). Such peer-led programs are highly desirable to agencies working with homeless youth as these youth are often disengaged from traditional health care settings and are distrustful of adults [Rice and Rhoades, 2013, Rice, 2010].

Agencies working with homeless youth prefer a series of small size interventions deployed sequentially as they have limited manpower to direct towards these programs. This fact, along with emotional and behavioral problems of youth makes managing groups of more than 5-6 youth at a time very difficult [Rice et al., 2012]. Strategically choosing intervention participants is important so that information percolates through their social network in the most efficient way.

In this chapter, I will use PSINET to study ranked voting aggregation. PSINET (**P**OMDP based **S**ocial **I**nterventions in **N**etworks for **E**nhanced HIV **T**reatment), is a Partially Observable Markov Decision Process (POMDP) based system which chooses the participants of successive interventions in a social network [Yadav et al., 2015]. The main motivation of PSINET is to choose participants for a sequence of interventions, where they would be educated about HIV-prevention, in order to spread that knowledge across the social network. As I will explain in detail in Section 6.4, PSINET considers the uncertainty over the edges (i.e., friendship connections) of the social network. Such uncertainty is very common when handling real life social networks, such as homeless populations.

6.2 Team Generation

I use a novel methodology for generating large teams. It is fundamentally different from that of Chapters 3 and 4, where I created a diverse team by combining four different, independently developed Go programs. Here I automatically create arbitrarily many diverse agents by parametrizing one Go program. Specifically, I use different parametrizations of Fuego 1.1 [Enzenberger et al., 2010]. Fuego is a state-of-the-art, open source, publicly available Go program; it won first place in 19×19 Go in the Fourth Computer Go UEC Cup, 2010, and also won first place in 9×9 Go in the 14th Computer Olympiad, 2009.

I sample random values for a set of parameters for each generated agent, in order to change its behavior. In Table 6.1 I present the parameters that were sampled to generate parametrized versions of Fuego. For each random draw, I used a uniform random distribution, defined in the interval shown in the column “Range”. Also, depending on the domain of each parameter, I sample integers or floating point numbers. A detailed description of these parameters is available in the Fuego documentation, at <http://fuego.sourceforge.net/fuego-doc-1.1/>.

Using this approach, I generate larger diverse teams than what I showed in previous chapters of this thesis. I will evaluate such teams in Section 6.5.1.

6.3 Ranked Voting

6.3.1 Ranking Extraction

Fuego (and, in general, all programs using Monte Carlo tree search algorithms) is not originally designed to output a ranking over all possible moves (alternatives), but rather to output a single move — the best one according to its search tree (there is no guarantee that the selected move is in fact the best one). Hence, we need to study ways to obtain a ranking from the agents. I study two different methodologies.

First, I modified Fuego to make it directly output a ranking over moves. When asked to find the best move, Fuego builds a search tree. Each node of the tree corresponds to a move in the current board state (given by the current level in the tree). In order to estimate the value of each node, Fuego runs simulations. Hence, each node has two values associated with it: p and n , where p is the probability of that move being the best one and n is the number of simulations used to estimate this probability. Due to the nature of Monte Carlo tree search algorithms, different nodes have wildly different n values. Hence, the comparison of moves according to the p values is unstable, and by default Fuego outputs the move with the highest n value. As a natural generalization,

Parameter	Domain	Range
uct_param_globalsearch mercy_rule	Integer	[0,1]
uct_param_globalsearch territory_statistics	Integer	[0,1]
uct_param_globalsearch length_modification	Float	[0, 0.5]
uct_param_globalsearch score_modification	Float	[0,0.5]
uct_param_player forced_opening_moves	Integer	[0,1]
uct_param_player reuse_subtree	Integer	[0,1]
uct_param_player use_root_filter	Integer	[0,1]
uct_param_policy nakade_heuristic	Integer	[0,1]
uct_param_policy fillboard_tries	Integer	[0, 5]
uct_param_rootfilter check_ladders	Integer	[0,1]
uct_param_search check_float_precision	Integer	[0,1]
uct_param_search prune_full_tree	Integer	[0,1]
uct_param_search rave	Integer	[0,1]
uct_param_search virtual_loss	Integer	[0,1]
uct_param_search weight_rave_updates	Integer	[0,1]
uct_param_search bias_term_constant	Float	[0, 1.0]
uct_param_search expand_threshold	Integer	[1,4]
uct_param_search first_play_urgency	Integer	[1,10000]
uct_param_search knowledge_threshold	Integer	[0,10000]
uct_param_search number_playouts	Integer	[1,3]
uct_param_search prune_min_count	Integer	[1,128]
uct_param_search randomize_rave_frequency	Integer	[0,200]
uct_param_search rave_weight_final	Integer	[1000,10000]
uct_param_search rave_weight_initial	Integer	[0,999]

Table 6.1: Parameters sampled to generate different versions of Fuego.

I inspect the first level of the final search tree, and rank the moves according to their n values.

I also introduce a novel ranking methodology, which I call “ranking by sampling”. For each board state, I sample moves from each agent, and rank the moves according to how frequently they were played by each agent. I generate the samples by repeatedly asking the agents where they would play, but without actually changing the current board state (I do not let the agents reuse parts of previous search trees, for the samples to be as independent as possible). In Section 6.5.2, I compare these two ranking procedures.

6.3.2 Ranked Voting Rules

In this chapter I will study the performance of 5 different ranked voting rules: Plurality, Borda, Harmonic, Maximin, and Copeland. For completeness, I briefly introduce those below:

- **Plurality:** Plurality is the simplest voting rule: given one vote per agent, the action with the highest number of votes is taken by the team. If the agents submit rankings, the plurality voting rule will consider only the top position of the rankings.
- **Borda:** The Borda voting rule assigns a score for each position in the ranking of the agents. Given a set of rankings (one per agent), the total score obtained by each action is calculated (by summing up the scores assigned for the action at each ranking – according to the position of the action at each ranking). The action with the highest total score is taken by the team. Traditionally, the scoring vector is $(m, m - 1, \dots, 1)$, where m is the size of the ranking. That is, at each ranking, the top action receives score m , the second top action score $m - 1$, and so on, until reaching the final action, which receives a score of 1. In this chapter I limit Borda to the top 6 positions in the rankings.
- **Harmonic:** The harmonic rule is similar to Borda: each position in the ranking is assigned a score, and rankings are aggregated by summing up the scores of each action. However, the scoring vector of the harmonic rule is defined as: $(1, 1/2, \dots, 1/m)$ [Boutilier et al., 2012].
- **Maximin:** Maximin follows a different approach than the previous rules. Instead of giving scores to each ranking position, it considers a pairwise comparison among all actions. Let $n(a_i, a_j)$ be the number of agents that rank action a_i higher than a_j . The maximin score of each action is its worst score: $\min_{i \neq j} n(a_i, a_j)$. The final result ranks the actions by their maximin score (i.e., the top action has the highest maximin score).

- **Copeland:** The Copeland rule follows a similar approach as the maximin rule: it considers a pairwise comparison among all actions. Again, let $n(a_i, a_j)$ be the number of agents that rank action a_i higher than a_j . An action a_i beats an action a_j if more agents rank a_i higher than a_j . That is, if $n(a_i, a_j) > n(a_j, a_i)$. The Copeland score of an action a_i is defined as the number of actions that a_i beats. Again, the final result ranks the actions by their Copeland score (i.e., the top action has the highest Copeland score).

6.4 PSINET

I also study ranked voting for influence maximization, using PSINET [Yadav et al., 2015]. In this section I briefly describe the system. Additional details are available at Yadav et al. [2015].

PSINET was developed to solve the influence maximization problem under uncertainty. Formally, let $G := (V, E)$ be a graph with a set of nodes V and edges E . We perform η interventions, where at each intervention we pick x nodes. A node may be either influenced or uninfluenced. An uninfluenced node may change to influenced, but an influenced node will never change back to uninfluenced. Each time we pick a node for an intervention, it will change to influenced. When a node changes from uninfluenced to influenced, it will “spread” the influence to its neighbors with some probability. That is, at each edge e there is a probability p_e . If a node v_1 is influenced, and there is an edge $e = (v_1, v_2)$, the node v_2 will also change to influenced with probability p_e . Similarly, if v_2 changes to influenced, it will spread the influence to its neighbors by the same process. Our objective is to maximize the number of influenced nodes after η interventions.

Contrary to the traditional independent cascade model [Kempe et al., 2003], Yadav et al. [2015] consider that each edge also has an existence probability q_e . Hence, if an edge e actually exists (with probability q_e), the influence will spread in the edge with probability p_e . If the edge does not exist (with probability $1 - q_e$), influence will not spread. Additionally, unlike the traditional model [Kempe et al., 2003], Yadav et al. [2015] consider that an influenced node v_i may spread influence to its neighbors after each time step, instead of only at the moment when it changed to influenced. Moreover, each time a node v_i is selected for an intervention, we are able to observe its edges. That is, we will be able to know which edges connected to v_i actually exist, and which edges do not exist.

6.4.1 Aggregation Methodologies

PSINET works by generating multiple agents before each intervention. Each agent votes for one action (or a raking over actions), where in this context an action represents which subset of nodes the agent thinks should be called for an intervention. The system takes an action by aggregating the opinions of all agents. I will study three different aggregation methodologies, which will be described later in this section.

Each agent is created by sampling one possible graph instantiation. That is, for each edge e , we decide whether it will exist or not with probability q_e . After going through all edges, we will have one possible network, which will be given to an agent. The agent, then, will run simulations in its sampled network, in order to decide which action to recommend.

I study three different aggregation methodologies:

- **PSINET-S:** Uses the usual plurality voting, where the vote of each agent has equal weight.
- **PSINET-W:** Uses weighted plurality voting. Let m be the number of uncertain edges. The agent which removes x uncertain edges has a vote weight of:

$$W(x) := \begin{cases} x & \forall x \leq m/2 \\ m - x & \forall x > m/2 \end{cases}$$

This weighting scheme approximates the probabilities of occurrences of real world events by giving low weights to instances which removes either too few or too many uncertain edges, since those events are less likely to occur. Instances which remove $m/2$ uncertain edges get the highest weight, since that event is most likely.

- **PSINET-C:** Uses the Copeland voting rule, described in the previous section. In order to use this aggregation methodology, it is necessary to have a ranking for each agent. I use the same method as in the previous section: each agent is queried multiple times, and the actions are ranked according to how frequent they are chosen by the agent (the most frequent action is assigned to the top position of the ranking). The rankings of the team are then aggregated using the Copeland voting rule.

In Section 6.5.3 I perform an experimental evaluation of these aggregation methodologies.

6.5 Results

6.5.1 Team Generation

I will start by presenting the results in the Computer Go domain. All results were obtained by simulating 1000 9×9 Go games, in an HP dl165 with dual dodeca core, 2.33GHz processors and 48GB of RAM. I compare the winning rates of games played against a fixed opponent. In all games the system under evaluation plays as white, against the original Fuego playing as black. I evaluate two types of teams: *Diverse* is composed of different agents, and *Uniform* is composed of copies of a specific agent (with different random seeds). In order to study the performance of the uniform team, for each sample (which is an entire Go game) I construct a team consisting of copies of a randomly chosen agent from the diverse team. Hence, the results presented for Uniform are approximately the mean behavior of all possible uniform teams, given the set of agents in the diverse team. In all graphs, the error bars show 99% confidence intervals.

Figure 6.1 shows the winning rates of Diverse and Uniform for a varying number of agents using the plurality voting rule. The winning rates of both teams increase as the number of agents increases. Diverse and Uniform start with similar winning rates, around 35% with 2 agents and 40% with 5 agents, but with 25 agents Diverse reaches 57%, while Uniform only reaches 45.9%. The improvement of Diverse over Uniform is not statistically significant with 5 agents ($p = 0.5836$), but is highly statistically significant with 25 agents ($p = 8.592 \times 10^{-7}$). I perform linear regression on the winning rates of the two teams to compare their rates of improvement in performance as the number of agents increases. Linear regression (shown as the dotted lines in Figure 6.1) gives the function $y = 0.0094x + 0.3656$ for Diverse ($R^2 = 0.9206$, $p = 0.0024$) and $y = 0.0050x + 0.3542$ for Uniform ($R^2 = 0.8712$, $p = 0.0065$). In particular, the linear approximation for the winning rate of Diverse increases roughly twice as fast as the one for Uniform as the number of agents increases.

I now analyze the parametrized agents, in order to better understand the performance of the diverse team. First, I show that the original Fuego is stronger than the parametrized agents. I ran 1000 9×9 Go games, with the system under evaluation playing as white, against the original Fuego playing as black. In Figure 6.2 we can see the winning rate of Fuego and of each one of the parametrized agents. The original Fuego is the strongest agent (with $p < 0.01$ for all but 3 agents), having a winning rate close to 50%. The parametrized agents, on average, have a winning rate of 32.3% (std: 10.4%).

I also evaluate the diversity of a team of parametrized agents, by analyzing a sample of 10 parametrized agents. I use the metric proposed in Chapter 3, where diversity is defined as the average Hellinger Distance [Hellinger, 1909] between the probability

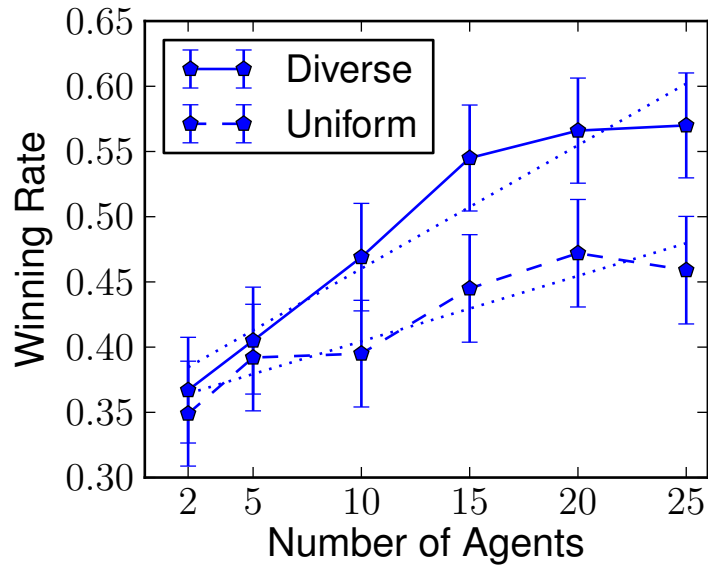


Figure 6.1: Winning rates for Diverse (continuous line) and Uniform (dashed line), for a variety of team sizes, using the plurality voting rule.

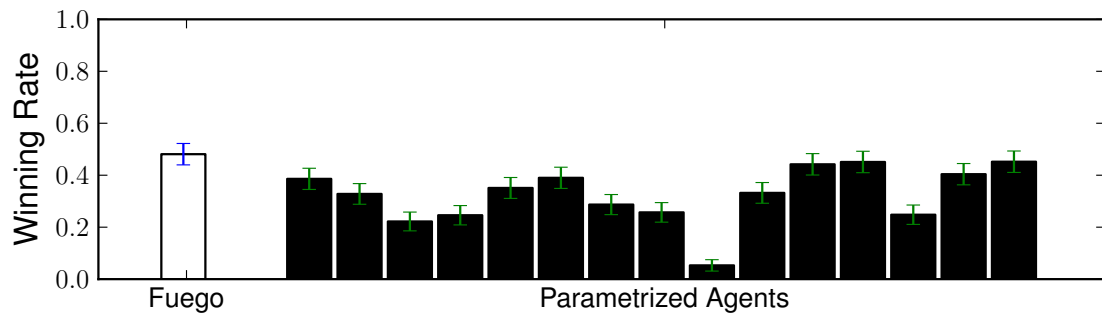


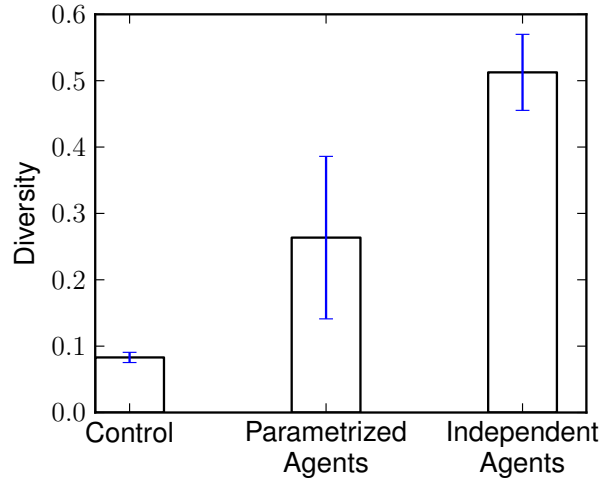
Figure 6.2: Winning rate of Fuego and of the parametrized agents.

distribution functions (pdfs) of all possible combinations of pairs of agents across different world states. I show three different results: *Control* compares each agent with a second sample of itself, in order to measure the noise in my evaluation; *Parametrized Agents* compares all possible pairs of parametrized agents, in order to estimate the diversity of our team; and *Independent Agents* compares each parametrized agent with Pachi [Baudiš and Gailly, 2011], an independently developed Computer Go program. In order to perform the analysis, I estimate the pdfs of Pachi and 10 agents from the diverse team, using 100 different board states. For each board state I sample 100 moves for each agent. The results are shown in Figure 6.3(a). These results indicate that the level of diversity is especially high when the parametrized agents are compared with Pachi, suggesting that the current parametrization methodology falls short of creating an idealized diverse team. That said, the methodology does lead to some diversity, as indicated by the statistically significant difference between the Control bar and Parametrized Agents bar.

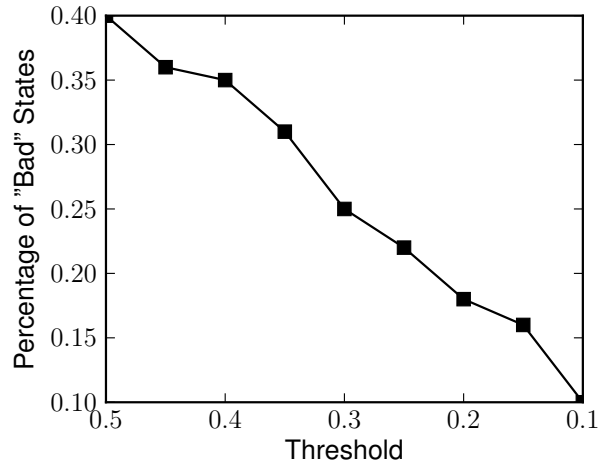
I also evaluate the level of diversity by testing whether there is a set of board states where all parametrized agents have a low probability of playing the best action. Again, I evaluate a sample of 10 agents from the diverse team. I first estimate the best move for each of 100 board states. To this end, I use Fuego to evaluate the given board state, but with a time limit 50x higher than the default one. Then, based on the previous estimated pdfs of the parametrized agents, we can obtain the probability of each agent playing the optimal action. Finally, I calculate the proportion of board states in which all parametrized agents play the best action with probability below a certain threshold. The results are shown in Figure 6.3(b). It turns out that all parametrized agents play the optimal action with probability smaller than $1/2$ in 40% of the board states. Moreover, in 10% of the board states, the probability of playing the optimal action is lower than 10%. Hence, there is still a large set of board states in which all agents play badly, regardless of the parametrization.

6.5.2 Ranked Voting

I first study the performance of the ranked voting rules, when extracting rankings directly from the search tree of the agents. Figure 6.4 compares the results across different voting rules. As mentioned, to generate ranked votes, I used the internal data in the search tree of an agent’s run (in particular, I rank using the number of simulations per alternative). We can see that increasing the number of agents has a positive impact for all voting rules under consideration. Moving from 5 to 15 agents for Diverse, plurality has a 14% increase in the winning rate, whereas other voting rules have a mean increase of only 6.85% ($std = 2.25\%$), close to half the improvement of plurality. For Uniform, the impact of increasing the number of agents is much smaller: Moving from 5 to 15 agents, the increase



(a) Diversity of the parametrized agents, compared with a second sample and with the diversity between independently developed agents



(b) Percentage of world states where all parametrized agents have probability of playing the best action below the given threshold

Figure 6.3: Evaluation of the diversity of the parametrized agents, and the fraction of states in which all of them have a low probability of playing the optimal action. The error bars show 99% confidence intervals.

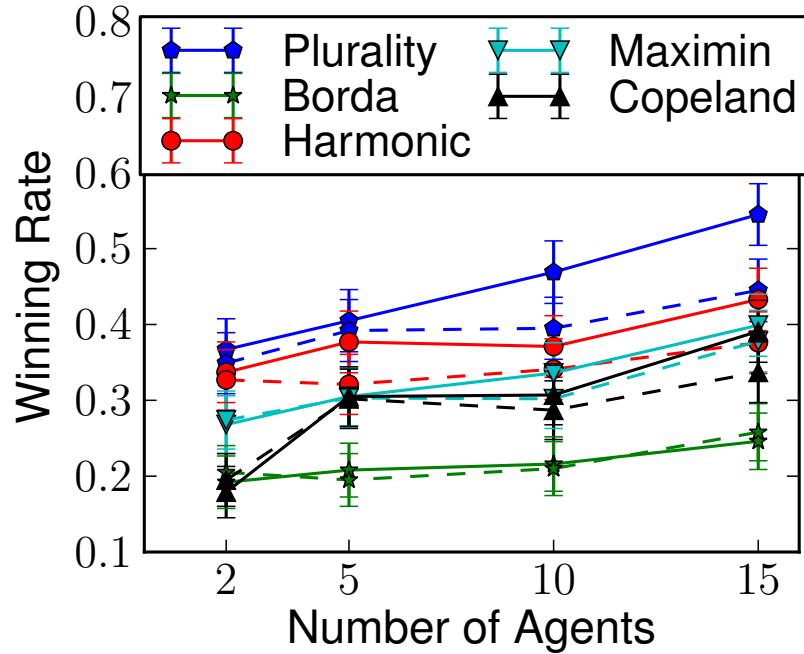


Figure 6.4: Winning rates for Diverse (continuous line) and Uniform (dashed line), for a variety of team sizes and voting rules.

for plurality is 5.3%, while the mean increase for other voting rules is 5.70% ($std = 1.45\%$). Plurality surprisingly seems to be the best voting rule in these experiments, even though it uses less information from the submitted rankings. This suggests that the ranking method used does not typically place good alternatives in high positions other than the very top.

Hence, I study now the performance of my novel ranking extraction methodology. As mentioned, to generate a ranked vote from an agent on a given board state, I run the agent on the board state 10 times (each run is independent of other runs), and rank the moves by the number of times they are played by the agent. I use these votes to compare plurality with the four other voting rules, for Diverse with 5 agents. Figure 6.5 shows the results. All voting rules outperform plurality; Borda and maximin are statistically significantly better ($p < 0.007$ and $p = 0.06$, respectively). All ranked voting rules are also statistically significantly better than the non-sampled (single run) version of plurality.

6.5.3 PSINET

I also evaluate aggregation methodologies for PSINET, in the context of influence maximization. I provide two sets of results. First, I show results on artificial networks to understand the algorithms' properties on abstract settings, and to gain insights on a

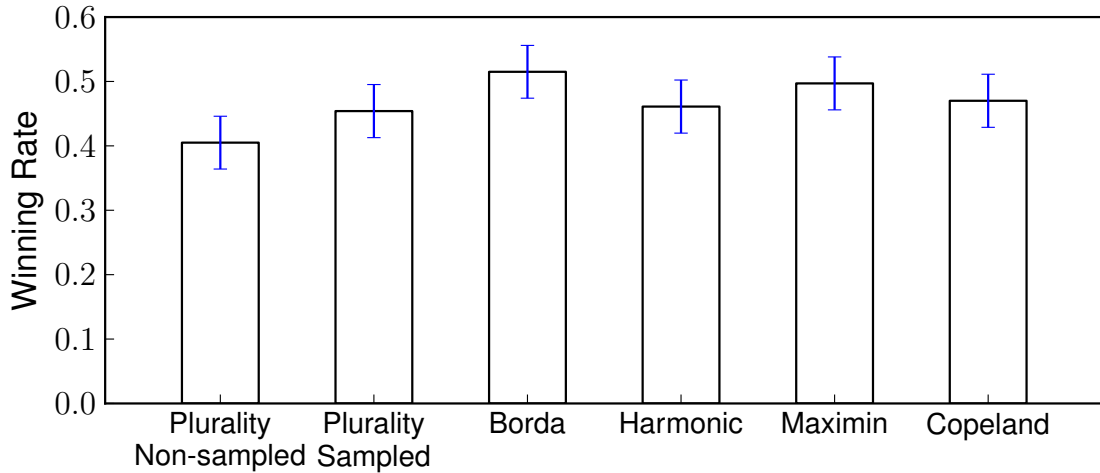


Figure 6.5: All voting rules, for Diverse with 5 agents, using the new ranking methodology.

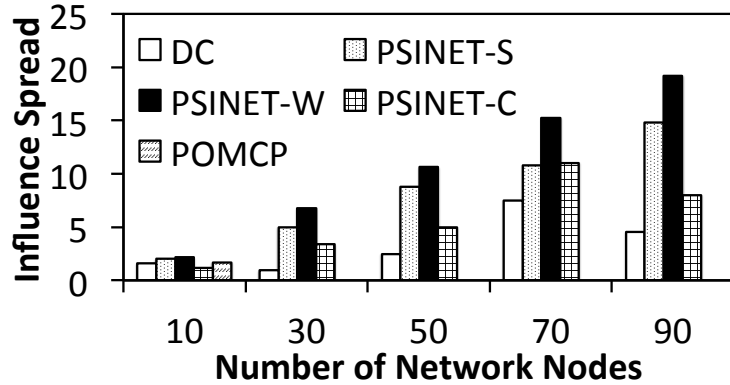


Figure 6.6: Comparison on BTER graphs

range of networks. Next, I show results on two real world homeless youth networks. In all experiments, I select 2 nodes per round and average over 20 runs. PSINET-S and PSINET-W use 20 agents (possible network instances) and PSINET-C uses 5 agents (each agent is sampled 5 times, in order to build a ranking). The propagation and existence probability values were set to 0.5 in all experiments (based on findings by Kelly et al. [1997]). In this section, a $\langle X, Y, Z \rangle$ network refers to a network with X nodes, Y certain and Z uncertain edges. I use a metric of “indirect influence spread” (IIS) throughout this section, which is the number of nodes “indirectly” influenced by the intervention participants. For example, on a 30 node network, by selecting 2 nodes each for 10 interventions (horizon), 20 nodes (a lower bound for any strategy) are influenced with certainty. However, the total number of influenced nodes might be 26 (say) and thus, the IIS is 6. All comparison results are statistically significant under bootstrap-t ($\alpha = 0.05$).

6.5.3.1 Artificial networks

First, I compare all algorithms on Block Two-Level Erdos-Renyi (BTER) networks (having degree distribution $X_d \propto d^{-1.2}$, where X_d is number of nodes of degree d) of several sizes, as they accurately capture observable properties of real-world social networks [Shadhri et al., 2012].

In Figure 6.6, I compare solution qualities of Degree Centrality (DC), POMCP [Silver and Veness, 2010] (since this problem can be modeled as a POMDP, see Yadav et al. [2015]), PSINET-S, PSINET-W and PSINET-C on BTER networks of varying sizes. In DC, nodes are selected in subsequent rounds in decreasing order of out-degrees, where every uncertain edge adds its existence probability to the node degrees. I choose DC as my baseline as it is the current modus operandi of agencies working with homeless youth. This figure shows that all algorithms beat DC by $\sim 60\%$. Further, it shows that PSINET-W beats PSINET-S and PSINET-C. Also, POMCP runs out of memory on 30 node graphs. Hence, as we can see, for PSINET plurality outperforms the Copeland ranked voting rule. However, weighted plurality obtains better results than simple plurality voting.

6.5.3.2 Real world networks

Figure 6.7 shows one of the two real-world friendship based social networks of homeless youth (created by my collaborators through surveys and interviews of homeless youth attending My Friend’s Place), where each numbered node represents a homeless youth. Figure 6.8 compares PSINET variants and DC on these two real-world social networks (each of size $\sim \langle 155, 120, 190 \rangle$). This figure clearly shows that all PSINET variants beat DC on both real world networks by $\sim 60\%$, which shows that PSINET works equally well on real-world networks. Also, PSINET-W (weighted plurality) beats PSINET-S (simple plurality), in accordance with previous results.

6.6 Conclusion

In this chapter I explored the automatic generation of agent teams, and ranked voting rules. I present results in two different domains: Computer Go and Social Networks. In Computer Go I show that we can improve performance with large agent teams, but the gain in performance decreases as the team grows. How to better generate diverse teams, in order to obtain even higher performance with large teams is still an open area for further study.

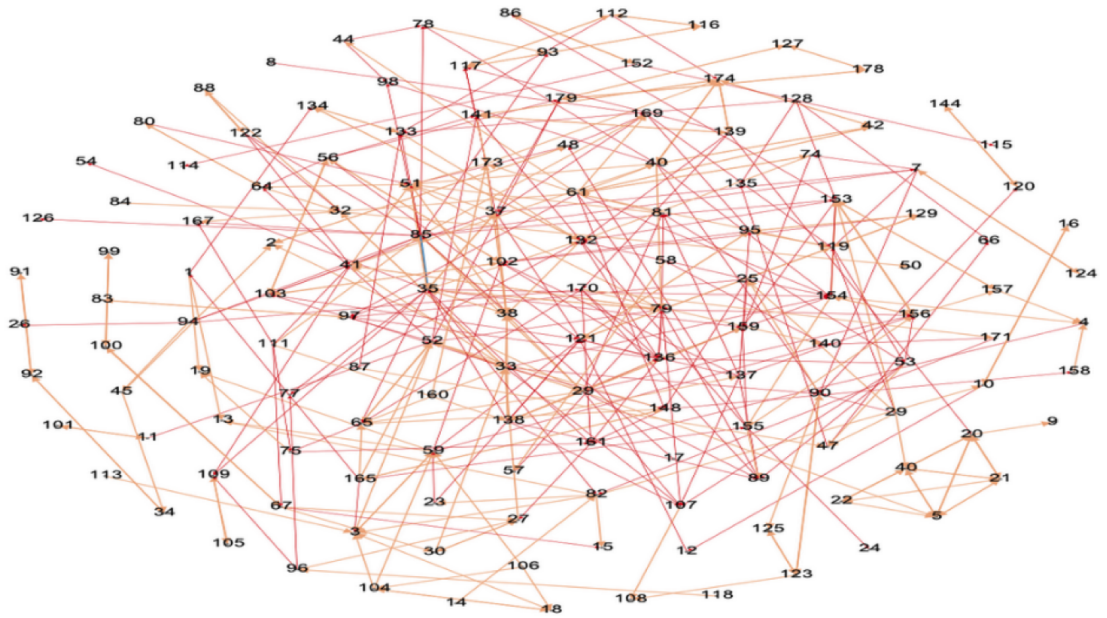


Figure 6.7: One of the friendship based social network of homeless people visiting My Friend's Place

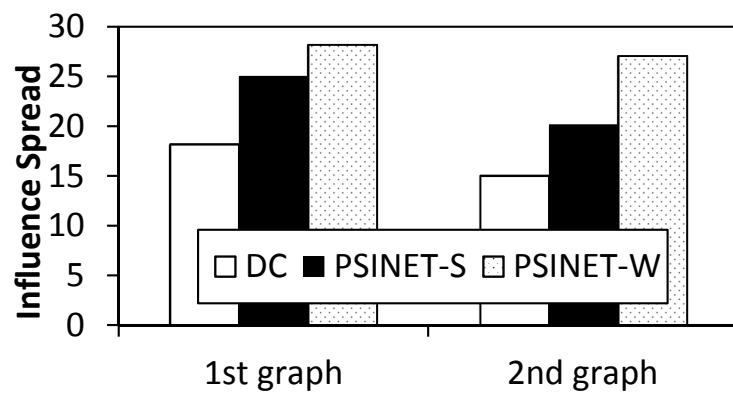


Figure 6.8: Solution Quality for Real World Networks

I also show that when directly extracting rankings from the search trees of the agents, the result of classical ranked voting rules is actually quite poor, and they are easily outperformed by simply playing plurality voting. However, it is possible to obtain better results by using a different ranking extraction technique, where rankings are built according to how frequently each action is played when an agent is sampled multiple times. When using this approach, I show that Borda outperforms plurality with statistical significance.

In the Social Networks domain, I study agents that vote to select subsets of nodes for influence maximization. I experimentally study the performance of plurality, weighted plurality and the Copeland ranked voting rule. In this domain plurality outperforms the Copeland voting rule; but I can obtain a better performance by using weighted plurality instead of using simple plurality voting.

Chapter 7

SIMULTANEOUS INFLUENCING AND MAPPING

*'Another may be sadly deceived
By the words you say.
And another, believing and trusting you,
May be led astray by the things you do.'*

*'For much that never you'll see or know
Will mark your days as you come and go.
And in countless lives that you'll never learn
The best and the worst of you will return.'*

(Edgar Albert Guest)

7.1 Introduction

Influencing a social network is an important technique, with great potential to positively impact society, as we can modify the behavior of a community. For example, we can increase the overall health of a population; Yadav et al. [2015], for instance, spread information about HIV prevention in homeless populations. However, although influence maximization has been extensively studied [Kempe et al., 2003, Cohen et al., 2014, Golovin and Krause, 2010], their main motivation is viral marketing, and hence they assume that the social network graph is fully known, generally taken from some social media network (such as Facebook).

However, the graphs recorded in social media do not really represent all the people and all the connections of a population. Most critically, when performing interventions in real life, we deal with large degrees of lack of knowledge. Normally the social agencies have to perform several interviews in order to learn the social network graph [Marsden, 2005].

These highly unknown networks, however, are exactly the ones we need to influence in order to have a positive impact in the real world, beyond product advertisement.

Additionally, learning a social network graph is very valuable *per se*. Agencies also need data about a population, in order to perform future actions to enhance their well-being, and better actuate in their practices [Marsden, 2005]. As mentioned, however, the works in influence maximization are currently ignoring this problem.

Each person in a social network actually knows other people, including the ones she cannot directly influence. Hence, each time we select someone for an intervention (to spread influence), we also have an opportunity to obtain knowledge from that person. Therefore, in this chapter I present for the first time the problem of simultaneously influencing and mapping a social network. I study the performance of the classical greedy influence maximization algorithm in this context, and show that it can be arbitrarily low. Hence, I study a class of algorithms for this problem, and show that we can effectively influence and map a network when *independence of objectives* holds. For the interventions where it does not hold, I give an upper bound in our loss, which converges to 0. I study an approximation of my main algorithm, that works as well but requiring fewer assumptions.

I perform a large scale experimentation using four real life social networks of homeless populations, where I show that my algorithm is competitive with previous approaches in terms of influence (even outperforming them in hard cases), and is significantly better in terms of mapping.

This chapter is the only one of the thesis that does not consider voting. Here the aggregation will be performed by a linear combination of two greedy algorithms, in order to simultaneously influence and map a social network.

7.2 Influencing and Mapping

I consider the problem of maximizing the influence in a social network. However, we start by knowing only a subgraph of the social network. Each time we pick a node to influence, it may teach us about subgraphs of the network. Our objective is to spread influence, at the same time learning the network graph (i.e., mapping). I call this problem as “Simultaneous Influencing and Mapping” (SIAM). In this chapter, I consider a version of SIAM where we only need to map the nodes that compose the network. I assume that we always know all the edges between the nodes of the known subgraph. For clarity, I will define formally here only the version of SIAM that I handle in this chapter. Therefore, unless otherwise noted, henceforth by SIAM I mean the version of the problem that is formally defined below.

Let $G := (V, E)$ be a graph with a set of nodes V and edges E . We perform η interventions, where at each one we pick one node. The selected node is used to spread influence and map the network. I assume we do not know the graph G , we only know a subgraph $G_k = (V_k, E_k) \subset G$, where k is the current intervention number. G_k starts as $G_k := G_0 \subset G$. For each node v_i , there is a subset of nodes $V^i \subset V$, which will be called “teaching list”. Each time we pick a node v_i , the known subgraph changes to $G_k := (V_{k-1} \cup V^i, E_k)$, where E_k contains all edges between the set of nodes $V_{k-1} \cup V^i$ in G . Our objective is to maximize $|V_k|$, given η interventions.

For each node v_i , I assume we can observe a number γ_i , which indicates the size of its teaching list. I study two versions: in one γ_i is the number of nodes in V^i that are not yet in G_k (hence, the number of new nodes that will be learned when picking v_i). I refer to this version as “perfect knowledge”. In the other, $\gamma_i := |V^i|$, and thus we cannot know how many nodes in V^i are going to be new or intersect with already known nodes in V_k . I refer to this version as “partial knowledge”. The *partial knowledge* version is more realistic, as by previous experience we may have estimations of how many people a person with a certain profile usually knows. I study both versions in order to analyze how much we may lose with *partial knowledge*. Note that we may also have nodes with empty teaching lists ($\gamma_i = 0$). The teaching list of a node v_i is the set of nodes that v_i will teach us about once picked, and is not necessarily as complete as the true set of all nodes known by v_i . Some nodes could simply refuse to provide any information.

Additionally, note that I am assuming the teaching list and the neighbor list to be independent. That is, a node may teach us about nodes that it is not able to directly influence. For instance, it is common to know people that we do not have direct contact with, or we are not “close” enough to be able to influence. Similarly, a person may not tell us about all her close friends, due to limitations of an interview process, or even “shame” to describe some connections. However, some readers could argue that people would be more likely to teach us about their direct connections. Hence, I will handle the case where the independence does not hold in my empirical experiments in Section 7.3.2.

Simultaneously to the problem of mapping, we also want to maximize the spread of influence over the network. I consider here the traditional independent cascade model, with observation, as in Golovin and Krause [2010]. That is, a node may be either influenced or uninfluenced. An uninfluenced node may change to influenced, but an influenced node will never change back to uninfluenced. Each time we pick a node for an intervention, it will change to influenced. When a node changes from uninfluenced to influenced, it will “spread” the influence to its neighbors with some probability. That is, at each edge e there is a probability p_e . When a node v_1 changes to influenced, if there is an edge $e = (v_1, v_2)$, the node v_2 will also change to influenced with probability p_e . Similarly, if

v_2 changes to influenced, it will spread the influence to its neighbors by the same process. Influence only spreads in the moment a node changes from uninfluenced to influenced. That is, a node may only “try” one time to spread influence to its neighbors. As in Golovin and Krause [2010], I consider that we have knowledge about whether a node is influenced or not (but in my case, we can only know about nodes in the current known subgraph G_k). Let I_k be the number of influenced nodes after k interventions. Our objective is to maximize I_k given η interventions. Influence may spread beyond G_k . Hence, I consider I_k as the number of influenced nodes in the full graph G . I denote as σ_i the expected number of nodes that will be influenced when picking v_i (usually calculated by simulations).

As mentioned, we want to attend both objectives simultaneously. Hence, we must maximize both $|V_k|$ and I_k . It is easy to show that SIAM is an NP-Complete problem:

Proposition 7.2.1 *SIAM is NP-Complete.*

Proof: Let κ be an instance of the influence maximization problem, with graph G . Consider now a SIAM problem where no node carries information and $G_0 := G$. If we can solve this SIAM problem we can also solve the influence maximization problem κ . Therefore, SIAM is NP-Complete. ■

As SIAM is NP-Complete, similarly to previous influence maximization works [Kempe et al., 2003, Golovin and Krause, 2010], I study greedy solutions. Like the exploration vs exploitation dilemmas in online learning [Valizadegan et al., 2011], the fundamental problem of SIAM is whether to focus on influencing or mapping the network. Hence, I propose as a general framework to select the node v_i such that:

$$v_i = \operatorname{argmax}(c_1 \times \sigma_i + c_2 \times \gamma_i) \tag{7.1}$$

Constants c_1 and c_2 control the balance between influencing or mapping. $c_1 = 1, c_2 = 0$ is the classical influence maximization algorithm (“*influence-greedy*”); $c_1 = 0, c_2 = 1$, on the other hand, only maximizes the knowledge-gain at each intervention (“*knowledge-greedy*”). $c_1 = c_2 = 1$ is an algorithm where both objectives are equally balanced (“*balanced*”). Different weights may also be used.

Remember that I defined two versions for the γ values: *perfect knowledge*, where we know how many new nodes a node will teach us about; and *partial knowledge*, where we do not know how many nodes will be new. In order to better handle the *partial knowledge* case, I also propose the “*balanced-decreasing*” algorithm, where c_2 constantly decreases until reaching 0. Hence, I define c_2 as:

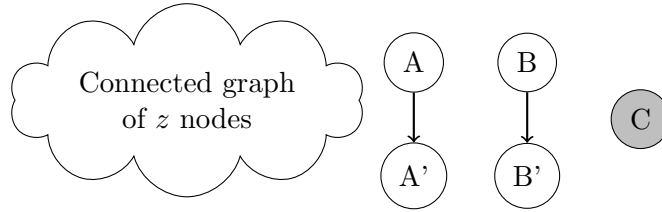


Figure 7.1: A graph where the traditional greedy algorithm has arbitrarily low performance.

$$c_2 := \begin{cases} c'_2 - \frac{1}{d} \times c'_2 \times k & \text{if } k \leq d \\ 0 & \text{otherwise} \end{cases}, \quad (7.2)$$

where c'_2 is the desired value for c_2 at the very first iteration, and d controls how fast c_2 decays to 0.

7.2.1 Analysis

I begin by studying *influence-greedy*. It was shown that when picking the node v which $\text{argmax}(\sigma_v)$ at each intervention, we achieve a solution that is a $(1 - 1/e - \epsilon)$ approximation of the optimal solution, as long as our estimation of σ_v (by running simulations) is “good enough” [Kempe et al., 2003]. However, even though the actual influence spread may go beyond the known graph G_k , we can only run simulations to estimate σ_v in the current G_k . Hence, the previous results are no longer valid. In fact, in the next observation I show that we can obtain arbitrarily low-performing solutions by using *influence-greedy*.

Observation 7.2.2 *The performance of influence-greedy can be arbitrarily low in a SIAM problem.*

I show with an example. Consider the graph in Figure 7.1, and assume we will run 2 interventions (i.e., pick 2 nodes). There is a probability 1 to spread influence in any edge. Our initial knowledge is $V_0 = \{A, A', B, B', C\}$. A and B can influence A' and B', respectively. However, C cannot influence any node. A, B, A' and B' have empty teaching lists. C, on the other hand, can teach us about a connected graph of z nodes. *Influence-greedy*, by running simulations on the known graph, picks nodes A and B, since each can influence one more node. The optimal solution, however, is to pick node C, which will teach us about the connected graph of z nodes. Then, we can pick one node in that graph, and influence $z + 1$ nodes in total. Hence, the *influence-greedy* solution is only $\frac{4}{z+1}$ of the optimal. As z grows, *influence-greedy* will be arbitrarily far from the optimal solution.

If we make some assumptions about the distribution of the teaching lists across the nodes, however, *influence-greedy* eventually maps the full graph given enough interventions. Let $n = |V|$, and $n_k = |V_k|$. I show the expected number of interventions to learn all n nodes (subtracted by a small ϵ , for numerical stability). I study the *partial knowledge* version. Assume the size of the teaching list of each node is drawn from a uniform distribution on the interval $[0, u]$, and any node is equally likely to be in a teaching list. I consider that there is a probability φ that a node will have a non-empty teaching list.

Proposition 7.2.3 *The expected number (k_{full}) of interventions for influence-greedy to learn $n - \epsilon$ nodes is $\frac{\log(-\frac{\epsilon}{n_0 - n})}{\log(1 - \frac{\varphi \times u}{2 \times n})}$.*

Proof: Since *influence-greedy* is not considering γ , it picks nodes arbitrarily in terms of knowledge-gain. Hence, on average it selects the expected value of the uniform distribution, $u/2$. For each node v in a teaching list, the probability that it is not yet known is $\frac{n - n_k}{n}$. Therefore, the expected number of nodes known at one iteration k is: $E[n_k] = \varphi \times \frac{u}{2} \times \frac{n - E[n_{k-1}]}{n} + E[n_{k-1}]$. Solving the recurrence gives: $E[n_k] = n_0 \times (1 - \frac{\varphi \times u}{2 \times n})^k - n \times (1 - \frac{\varphi \times u}{2 \times n})^k + n$. Solving for $E[n_k] = n - \epsilon$ gives that the expected number of interventions is: $k_{full} = \frac{\log(-\frac{\epsilon}{n_0 - n})}{\log(1 - \frac{\varphi \times u}{2 \times n})}$. ■

k_{full} quickly increases as φ (or u) decreases. In Section 7.3, I study experimentally the impact of φ on the performance of *influence-greedy*.

Now, let's look at *balanced*. Clearly, it will learn the full graph with a lower number of expected interventions than *influence-greedy*. However, although intuitively *balanced* may seem reasonable, its performance may also quickly degrade if we assume *partial knowledge* (i.e., $\gamma_i = |V^i|$).

Proposition 7.2.4 *The performance of the balanced algorithm degrades as $n_k \rightarrow n$, if $\gamma_i = |V^i|$.*

Proof: Each node in the teaching list of a node v_i has probability $\frac{n - n_k}{n}$ of being a yet unknown node. Hence, the expected number of unknown nodes that will be learned by picking a node with teaching list size γ_i is: $E[new] = \gamma_i \times \frac{n - n_k}{n}$. As $n_k \rightarrow n$, $E[new] \rightarrow 0$. Hence, when $n_k \rightarrow n$, *balanced* picks a node v that maximizes $\sigma_v + \gamma_v$, thus missing to select nodes v_o (if available) with $\sigma_o > \sigma_v$, $\sigma_o + \gamma_o < \sigma_v + \gamma_v$, with no actual gains in mapping. ■

This problem does not happen in the *perfect knowledge* version. Since the γ values only include new nodes, $\gamma \rightarrow 0$ as $n_k \rightarrow n$, for all γ . Hence, in the *perfect knowledge* version, *balanced* converges to the same behavior as *influence-greedy* as k increases. In

order to approximate this behavior for the *partial knowledge* case, I propose the *balanced-decreasing* algorithm, where the constantly decreasing c_2 “simulates” the decreasing γ values.

I now show that *balanced* can match the performance of *influence-greedy* in terms of influence, but at the same time mapping better the network. As I just discussed that *perfect knowledge* can be approximated by using *balanced-decreasing*, I focus here on the *perfect knowledge* case. I show that when the *independence of objectives* hypothesis holds (defined below), *balanced* plays the same as *influence-greedy* or better, while *influence-greedy* may still fail in terms of mapping. If the hypothesis does not hold, our influence loss at one intervention will be bounded by $u/2 \rightarrow 0$.

Let V_k^σ be a subset of V_k where each $v \in V_k^\sigma$ maximizes σ in the current intervention k . Similarly, let $V_k^\gamma \subset V_k$, where each $v \in V_k^\gamma$ maximizes γ in the current intervention k . As before, I consider that the teaching list size of a node is given by a uniform distribution, but since γ tends to decrease at each intervention, I denote the interval as $[0, u_k]$.

Clearly, any node in the set $V_k^{Good} := V_k^\sigma \cap V_k^\gamma$ should be selected, as they maximize both objectives. Hence, when $V_k^{Good} \neq \emptyset$ it is possible to simultaneously maximize both objectives, and thus we say that the *independence of objectives* hypothesis holds. Since we are studying greedy-algorithms, both *balanced* and *influence-greedy* lack optimality guarantees. Hence, I focus here on a “local” analysis, and show that given a set of k possible interventions (with the same graph state across both algorithms at each intervention), *balanced* is able to pick nodes that spread as much influence as *influence-greedy*. Moreover, when *balanced* picks a different node, our loss is bounded by $u_k/2$. As $u_k \rightarrow 0$ with $k \rightarrow \infty$, our loss also converges to 0.

Proposition 7.2.5 *Balanced selects nodes that spread as much influence as influence-greedy, if $|V_k^\sigma| > n_k/2$ and $|V_k^\gamma| > n_k/2$, or as $k \rightarrow \infty$. Influence-greedy, on the other hand, selects worse nodes than balanced in terms of mapping with probability $1 - \frac{|V_k^\sigma \cap V_k^\gamma|}{|V_k^\sigma|}$. Moreover, when balanced selects a node with worse σ than influence-greedy, the expected influence loss is bounded by $u_k/2$, which $\rightarrow 0$ as $k \rightarrow \infty$.*

Proof: As *balanced* plays $\text{argmax}(\sigma + \gamma)$, if there is a node $v \in V_k^{Good}$, *balanced* picks v . *Influence-greedy*, however, selects an arbitrary node in V_k^σ . Hence, it picks a node $v \in V_k^{Good}$ with probability $\frac{|V_k^\sigma \cap V_k^\gamma|}{|V_k^\sigma|}$. Therefore, for all interventions where $V_k^{Good} \neq \emptyset$, *balanced* selects a node in V_k^{Good} , while *influence-greedy* makes a mistake in terms of mapping with probability $1 - \frac{|V_k^\sigma \cap V_k^\gamma|}{|V_k^\sigma|}$.

We consider now the probability of $V_k^{Good} \neq \emptyset$ across k interventions. Clearly, if $|V_k^\sigma| > n_k/2$, and $|V_k^\gamma| > n_k/2$, we have $V_k^{Good} \neq \emptyset$. If not, note that as $k \rightarrow \infty$, $n_k \rightarrow n$.

Therefore, $V_k^\gamma \rightarrow V_k$ (since all $\gamma \rightarrow 0$, all nodes will have the same teaching list size), thus $V_k^{Good} \rightarrow V_k^\sigma \neq \emptyset$. Hence, the probability of $V_k^{Good} \neq \emptyset$ goes to 1 as $k \rightarrow \infty$.

Let's study now the case when $V_k^{Good} = \emptyset$. Let v_1 be the node in V_k^σ picked by *influence-greedy*, and v_2 be the node in $V_k \setminus V_k^\sigma$ with the largest γ_2 . Since $V_k^{Good} = \emptyset$, we must have that $\sigma_1 > \sigma_2$, and $\gamma_2 > \gamma_1$. However, as long as $\gamma_2 - \gamma_1 < \sigma_1 - \sigma_2$, *balanced* still selects v_1 (or an even better node). In the worst case, the expected value for γ_2 is the expected maximum of the uniform distribution: $E[\gamma_2] = u_k - u_k/(n_k + 1) \leq u_k$. γ_1 , on the other hand, has the expected value of the uniform distribution $E[\gamma_1] = u_k/2$. Hence, as long as $\sigma_1 - \sigma_2 > u_k/2$, in expectation *balanced* still picks v_1 (or an even better node). Moreover, when *balanced* does not pick v_1 , our loss in terms of influence at intervention k is at most $u_k/2$. Since $\gamma \rightarrow 0$ as $n_k \rightarrow n$, $u_k/2 \rightarrow 0$ as $k \rightarrow \infty$. ■

Proposition 7.2.5 shows that we may experience loss in one intervention, when comparing *influence-greedy* with *balanced*. However, the loss is bounded by $u_k/2$, which goes to 0 as the number of interventions grows. Moreover, when we do not update the γ values, we can use the *balanced-decreasing* algorithm to simulate the same effect. Additionally, in Proposition 7.2.5 I considered the same graph states at each intervention across both algorithms. In practice, however, since *balanced* is able to map the network faster, any loss experienced in the beginning when k is low can be compensated by playing better later with full knowledge of the graph, while *influence-greedy* may still select nodes with lower σ due to lack of knowledge. As noted in Observation 7.2.2, lack of knowledge of the full graph can make *influence-greedy* play with arbitrarily low performance. In Section 7.3.2 I perform an empirical analysis assuming a power law model for the teaching lists, and I note here that my main results still hold.

7.3 Results

I run experiments using four real life social networks of the homeless population of Los Angeles, provided by Eric Rice, from the School of Social Work of the University of Southern California. All the networks are friendship-based social networks of homeless youth who visit a social agency. The first two networks (A, B) were created through surveys and interviews. The third and fourth networks (Facebook, MySpace) are online social networks of these youth created from their Facebook and MySpace profiles, respectively.

I run 100 executions per network. At the beginning of each execution, 4 nodes are randomly chosen to compose our initial subgraph (G_0). As mentioned, I consider that we always know the edges between the nodes of our current knowledge graph (G_k). I noticed similar tendencies in the results across all four social networks. For clarity, I plot here the results considering all networks simultaneously (that is, I average over all the

400 executions). In the Appendix A I show the individual results for each network. In all graphs, the error bars show the confidence interval, with $\rho = 0.01$. When I say that a result is significantly better than another, I mean with statistical significance according to a *t-test* with $\rho \leq 0.01$, unless noted otherwise. The size of each network is: 142, 188, 33, 105; for A, B, Facebook and MySpace, respectively. I evaluate up to 40 interventions.

I measure the percentage of influence in the network (“Influence”) and percentage of known nodes (“Knowledge”) for *influence-greedy*, *knowledge-greedy*, *balanced* and *balanced-decreasing* (with $c'_2 = 1.0$, and $d = 5$). In order to estimate the expected influence spread (σ_v) of each node, I run 1000 simulations before each intervention. Estimating the expected influence through simulations is a common method in the literature. In our case, the simulations are run in the current known subgraph G_k , although the actual influence may go beyond G_k . Influenced nodes in $G \setminus G_k$ will be considered when I measure Influence, but will not be considered in my estimation of σ_v . Concerning the teaching list size (γ_v), I consider it to hold the number of *new* nodes that would be learned if v is selected, for *balanced* and *knowledge-greedy* (i.e., perfect knowledge). For *balanced-decreasing*, I consider γ_v to hold the full teaching list size, including nodes that are already known (i.e., partial knowledge). Therefore, we can evaluate if *balanced-decreasing* approximates well *balanced*, when perfect knowledge is not available.

I simulate the teaching lists, since there are no real world data available yet (I only have data about the connections in the four real life social networks). I study two models: (i) uniform, which follows the assumptions of my theoretical analysis; (ii) power law, which considers that nodes are more likely to teach us about others which are close to them in the social network graph. I present the second model to show that my conclusions hold irrespective of the uniform assumption. For each node, we decide whether it will have a non-empty teaching list according to a probability φ . I run experiments using different combinations of φ , probability of influence p , and c_1 and c_2 values.

7.3.1 Uniform Model

Under the uniform model, if a node has a teaching list, I fix its size according to a uniform distribution from 0 to $0.5 \times |V|$. Each node in the graph is also equally likely to be in the teaching list of a node v_i . I consider here the teaching list and the neighbor list to be independent, as people may know others that they cannot influence, and they may also not tell us all their connections, as described before. The case where the teaching list and the neighbor list are not independent is considered in Section 7.3.2.

I run several parametrizations. Figure 7.2 shows the result at each intervention for $\varphi = 0.5$ and $p = 0.5$. As we see in Figure 7.2 (a), the Influence obtained by *influence-greedy*, *balanced*, and *balanced-decreasing* are very similar. In fact, out of all 40 interventions,

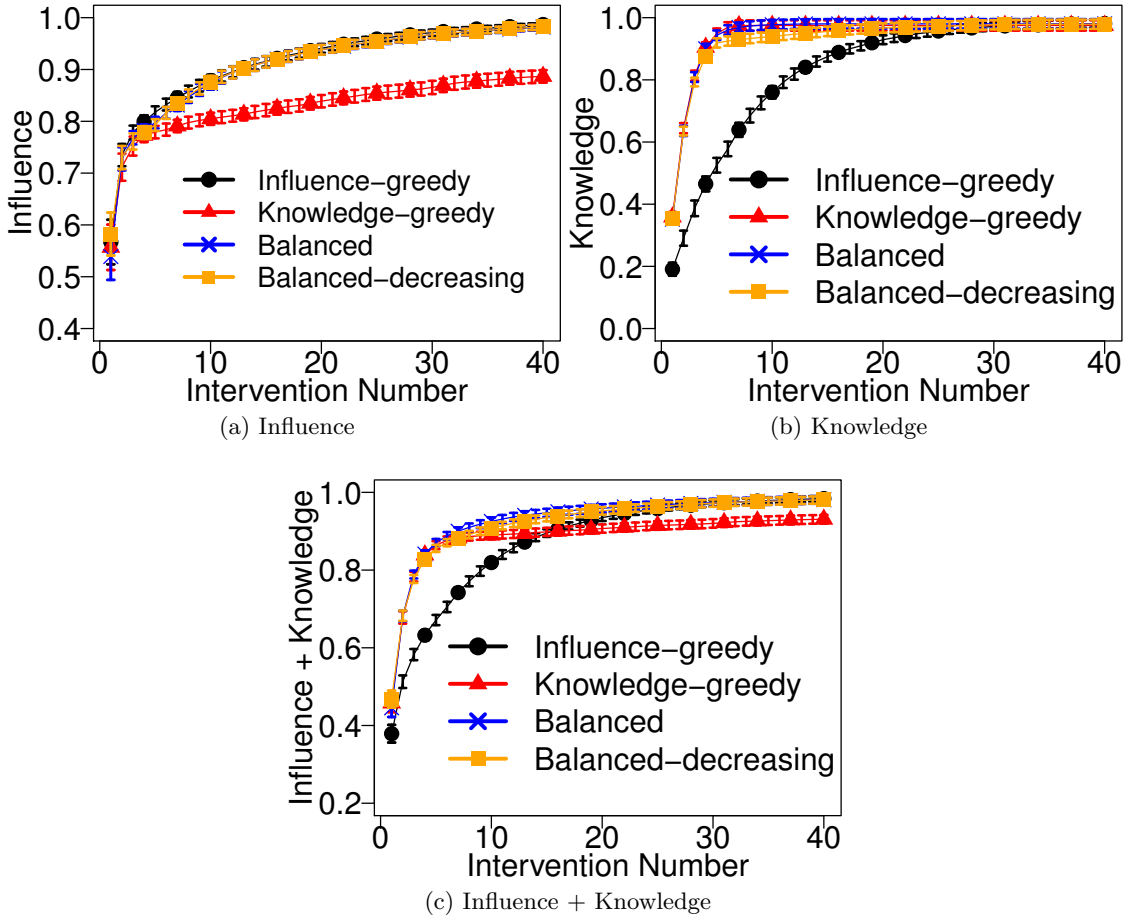


Figure 7.2: Results of 4 real world networks across many interventions, for $p = 0.5$ and $\varphi = 0.5$ (uniform distribution).

their result is not significantly different in any of them (and they are significantly better than *knowledge-greedy* in around 75% of the interventions). This shows that *balanced* is able to successfully spread influence in the network, while at the same time mapping the graph. We can also notice that a perfect knowledge about the number of new nodes in the teaching lists is not necessary, as *balanced-decreasing* obtained close results to *balanced*.

Figure 7.2 (b) shows the results in terms of Knowledge. All algorithms clearly outperform *influence-greedy* with statistical significance. Moreover, the result for *knowledge-greedy*, *balanced* and *balanced-decreasing* are not significantly different in any of the interventions. This shows that we are able to successfully map the network (as well as *knowledge-greedy*), but at the same time spreading influence successfully over the network (as well as *influence-greedy*), even in the *partial knowledge* case. Hence, the *independence of objectives* hypothesis seems to hold at most interventions in the networks, since we

could maximize both objectives simultaneously, as predicted in Proposition 7.2.5. Given enough interventions, however, *influence-greedy* is also able to map the network, as I discussed in Proposition 7.2.3.

It is also interesting to note that even though *influence-greedy* has much less information about the network (with significantly lower mapping performance in around 16 interventions), it is still able to perform as well as the other algorithms in terms of Influence. Observation 7.2.2, however, showed that its Influence performance can be arbitrarily low. As I discuss later, for some parametrizations we actually found that *influence-greedy* has significantly lower results than the other algorithms in terms of Influence as well.

In order to compare the results across different parametrizations, we calculate the area under the curve (AUC) of the graphs. The closer the curves are to 1.0 the better, hence an AUC of 39 (that is, always at 1.0 across all 40 interventions) would be an “ideal” result. In Figure 7.3 (a) I show the results for a fixed influence probability value ($p = 0.5$), but different teaching probability (φ) values. First I discuss the results in terms of Influence (left-hand side of the graph). As we can see, except for *knowledge-greedy*, all algorithms obtain very similar results. However, for $\varphi = 0.1$, the Influence for *balanced* and *balanced-decreasing* is slightly better than *influence-greedy*, in the borderline of statistical significance ($\rho = 0.101$ and 0.115 , respectively). Moreover, we can see that φ does impact the influence that we obtain over the network, although the impact is not big. For *influence-greedy*, from $\varphi = 0.5$ to $\varphi = 1.0$, the difference is only statistically significant with $\rho = 0.092$. However, from $\varphi = 0.1$ to $\varphi = 0.5$ the difference is statistically significant with $\rho = 3.26 \times 10^{-27}$. Similarly, for all other algorithms there is a significant difference from $\varphi = 0.1$ to $\varphi = 0.5$, while from $\varphi = 0.5$ to $\varphi = 1.0$ the difference is only significant with $\rho < 0.1$ (except for *knowledge-greedy*, its difference is not significant between $\varphi = 0.5$ and $\varphi = 1.0$).

Let’s look at the results in terms of Knowledge, on the right-hand side of Figure 7.3 (a). We can see that φ has a much bigger impact in our mapping, as expected. *Knowledge-greedy*, *balanced* and *balanced-decreasing* are all significantly better than *influence-greedy*. However, we can notice that the difference between *influence-greedy* and the other algorithms decreases as φ increases. Similarly, when comparing *knowledge-greedy*, *balanced* and *balanced-decreasing*, we can notice that the difference between the algorithms also decreases as φ increases. For both $\varphi = 0.1$ and $\varphi = 0.5$, however, the algorithms are not significantly different. Interestingly, when $\varphi = 1$, because of the lower variance, *knowledge-greedy* and *balanced* become significantly better than *balanced-decreasing*, even though the differences between the algorithms decreases.

In Figure 7.3 (b), I keep $\varphi = 0.5$, and change p . In the left-hand side we see the results for Influence. As expected, there is clearly a significant difference when p changes

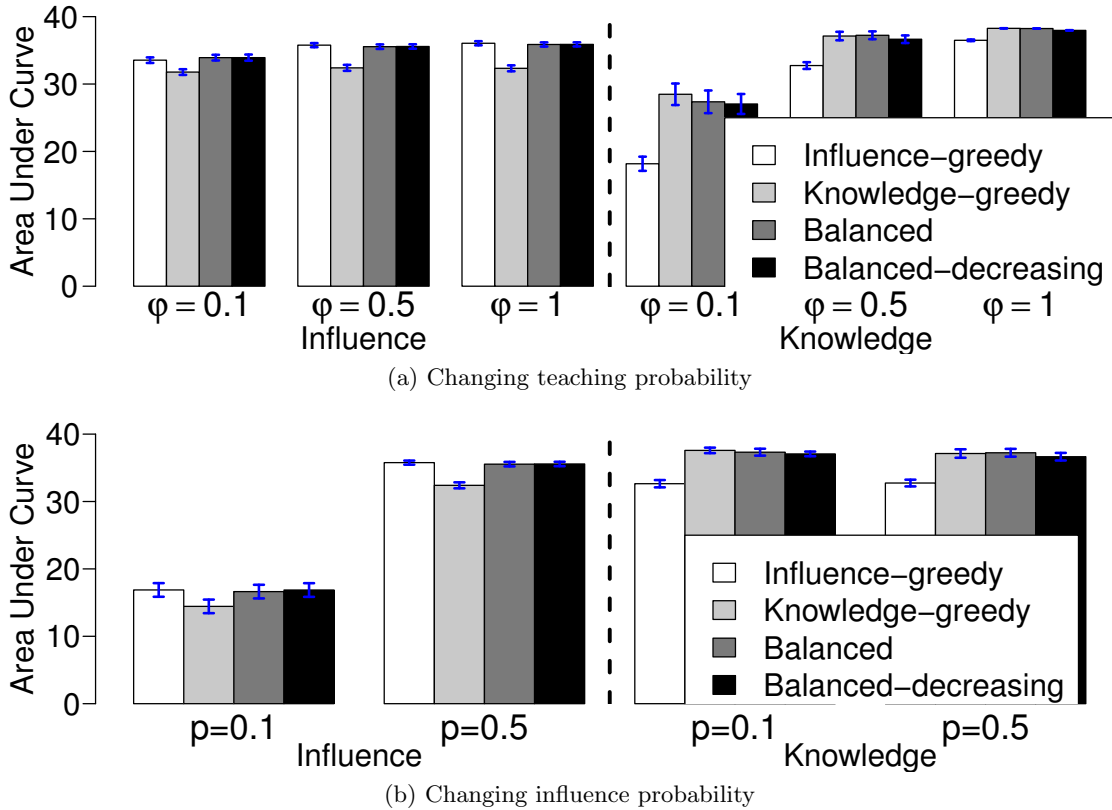


Figure 7.3: Results of Influence and Knowledge for different teaching and influence probabilities (uniform distribution).

from 0.1 to 0.5. However, we can notice that the difference between the algorithms does not change significantly when p changes. In both cases, the differences between *influence-greedy*, *balanced* and *balanced-decreasing* are not significant. Additionally, in both cases all algorithms are significantly better than *knowledge-greedy*. In terms of Knowledge (right-hand side of the figure) we see that the influence probability has no impact in any algorithm, as it would be expected. For all algorithms, the difference between $p = 0.1$ and $p = 0.5$ is not statistically significant.

I also compare the *regret* obtained by the different algorithms at different influence probabilities and teaching probability values. First, I run the *influence-greedy* algorithm, but considering that we know the full graph (that is, $G_k := G$). Although that solution is not optimal, it is the best known approximation of the optimal, hence I call it “perfect”. I calculate the AUC for *perfect*, and define the regret of an algorithm x as: $AUC_{Perfect} - AUC_x$. I analyze the regret in terms of Influence in Figure 7.4. Note that, in the figure, the lower the result the better. On the left-hand side I show the regret for $p = 0.1$ and different φ values, while on the right-hand side I show for $p = 0.5$. All algorithms

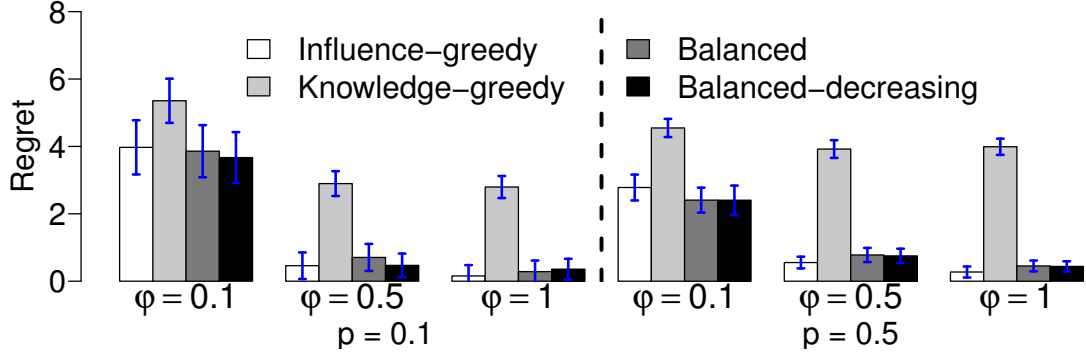


Figure 7.4: Regret for different teaching and influence probabilities (uniform distribution). Lower results are better.

(except *knowledge-greedy*) have a similar regret, as it would be expected based on the previous results. However, we can notice here that the regret for *balanced* and *balanced-decreasing* when $\varphi = 0.1$ and $p = 0.5$ is actually *lower* than *influence-greedy*. The difference is statistically significant, with $\rho = 0.069$ and $\rho = 0.092$, respectively. Hence, we can actually have a significantly better influence on the social network graph than the traditional greedy algorithm, when the teaching probability is low. However, when $\varphi = 0.1$ and $p = 0.1$, even though the regret for *balanced* and *balanced-decreasing* is still lower than *influence-greedy*, it is not significant anymore (as there is larger variance when $p = 0.1$). Additionally, we note that the difference in regret for *balanced* and *balanced-decreasing* is always not significant. It is also interesting to note that for some parametrizations, the regret of *influence-greedy* is actually close to 0, which means that in some cases lack of knowledge of the full graph does not significantly harm the influence performance. When $p = 0.1$, and $\varphi = 0.5$ or $\varphi = 1.0$, the regret is not significant ($\rho = 0.410$, and $\rho = 0.78$, respectively). For $p = 0.5$ and $\varphi = 1.0$, the regret is in the borderline of not being significant ($\rho = 0.102$). In all other cases, the regret is significant.

I discussed 4 algorithms, but my framework can actually generate a variety of behaviors by using different c_1 and c_2 values. I tested 6 more combinations: $\{(0.5, 2), (1, 0.5), (1, 1.5), (1, 2), (2, 0.5), (2, 1)\}$, but I did not observe significant differences in comparison with the previous algorithms in the four social networks graphs. Hence, finding a good parametrization of c_1 and c_2 does not seem to be a crucial problem for the *balanced* algorithm.

7.3.2 Power Law Model

In order to show that my conclusions still hold under different models, I also run experiments considering a power law distribution for the teaching lists. The power law is a very

suitable model for a range of real world phenomena. In fact, Andriani and McKelvey [2007], in a very comprehensive literature survey, lists 80 different kinds of phenomena which are modeled in the literature by power law distributions, and half of them are social phenomena. For example, it has been shown to be a good model for social networks, co-authorships networks, the structure of the world wide web and actor movie participation networks. A power law model also seems suitable in our case, as we can expect that a person will be very likely to teach us about the people who she has a direct connection with, and less and less likely to report people that are further away in the graph. Hence, when generating the teaching list of a node v_i in my experiments, each node v_o ($v_o \neq v_i$) will be in its teaching list according to the following probability: $p_o := (a - 1.0) \times h_o^{-a}$, where $1.0 < a \leq 2.0$, and h_o is the shortest path distance between node v_i and v_o . $a - 1.0$ represents the probability of a neighbor node v_o (i.e., $h_o = 1$) being selected. If node v_i and v_o are completely disconnected, I set $h_o := |V|$. Under this model the probability of a person teaching us about another is always strictly greater than 0, even though it may be very small if the respective nodes are very distant in the graph.

I fix $a = 1.8$ (80% probability of each of a nodes' neighbors being in its teaching list). I show results for $a = 1.2$ in Appendix A, for the interested reader (and my conclusions still hold in the alternative parametrization). Similarly as before, Figure 7.5 shows the result at each intervention for $\varphi = 0.5$ and $p = 0.5$. As we can see, my main conclusions still hold in the power law model. The Influence obtained by *influence-greedy*, *balanced*, and *balanced-decreasing* are very similar. Out of 40 interventions, their results are not significantly different in 39 ($\rho \leq 0.05$), and they are significantly better than *knowledge-greedy* in around 60% of the interventions ($\rho \leq 0.05$).

This time, however, *balanced-decreasing* obtained worse results than *balanced* in terms of Knowledge. Although up to iteration 4, *knowledge-greedy*, *balanced* and *balanced-decreasing* are not significantly different; both *knowledge-greedy* and *balanced* are significantly better than *balanced-decreasing* after that iteration. It is harder to obtain Knowledge under the power law model than under the uniform model (all algorithms converge slower to 1.0 than before). Hence, *balanced-decreasing* would require a slower decay speed (i.e., higher d) in this case, in order to perform better.

We can also notice that all algorithms are significantly better than *influence-greedy* in all iterations, in terms of Knowledge. Note that under the uniform model, *influence-greedy* was not significantly worse than the other algorithms after iteration 20 ($\rho \leq 0.1$). Hence, as expected, *influence-greedy* becomes relatively worse than the other algorithms when we assume a model where mapping is harder.

I calculate the AUC, in order to compare different parametrizations. Figure 7.6 (a) shows the result for a fixed influence probability value ($p = 0.5$), and different teaching

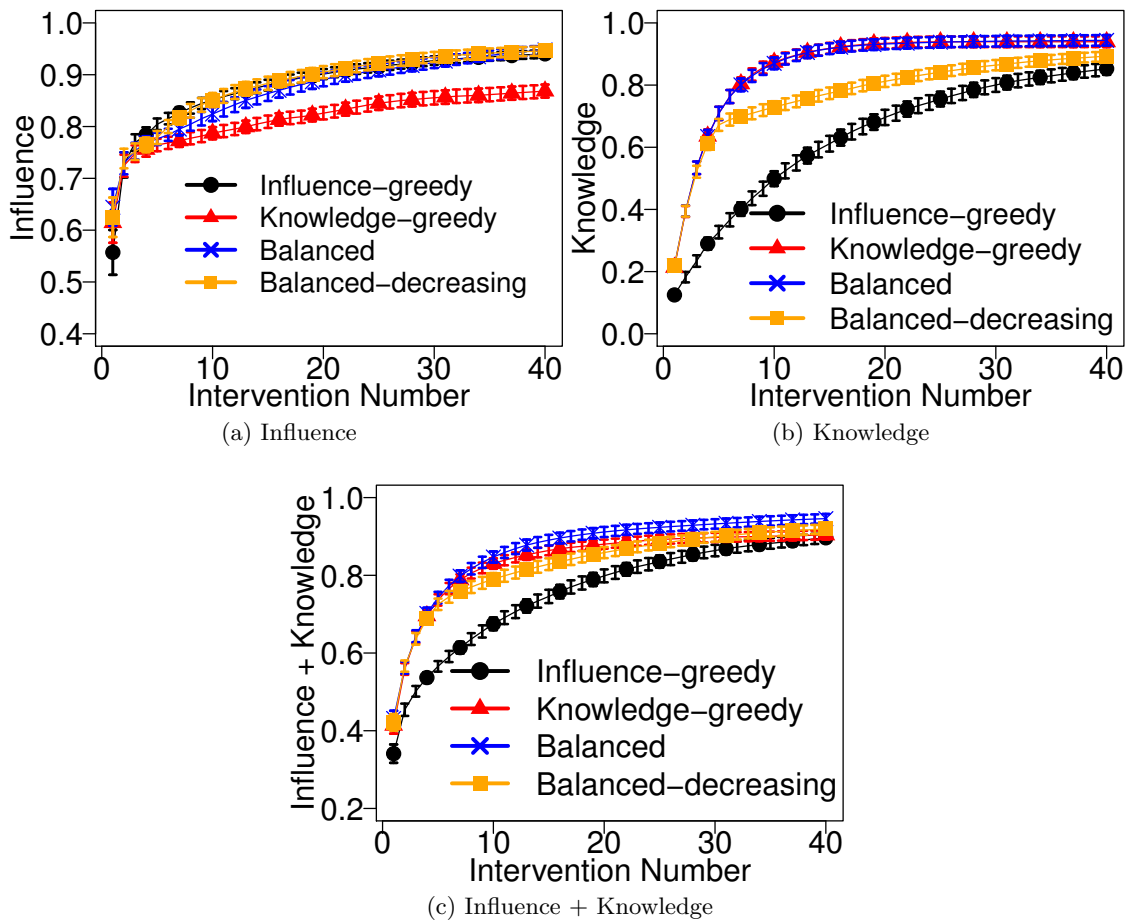


Figure 7.5: Results of 4 real world networks across many interventions, for $p = 0.5$ and $\varphi = 0.5$ (power law distribution).

probability (φ) values. As before, except for *knowledge-greedy*, all algorithms in general obtain similar results in terms of Influence. We notice, however, that for $\varphi = 0.1$, the result for *balanced* and *balanced-decreasing* is actually significantly better than *influence-greedy* ($\rho = 0.064$ and 0.038 , respectively). Again, we also notice that φ significantly impacts the influence that we obtain over the network, although the impact is small. For all algorithms, the impact is significant from $\varphi = 0.1$ to $\varphi = 0.5$. For *influence-greedy*, the difference is statistically significant with $\rho = 1.39 \times 10^{-31}$, while for *knowledge-greedy*, *balanced* and *balanced decreasing*, there is statistically significant difference with $\rho = 4.529 \times 10^{-6}$, 3.258×10^{-14} and 8.429×10^{-20} , respectively. However, the impact of φ increasing from 0.5 to 1 is not that significant for *knowledge-greedy*, *balanced* and *balanced-decreasing* ($\rho = 0.41, 0.02, 0.03$, respectively), while for *influence-greedy* the change has significant impact ($\rho = 10^{-6}$). In terms of Knowledge, we can see that all algorithms are significantly better than *influence-greedy* for all φ values (with $\rho \leq 3.464015 \times 10^{-19}$). However, this time we notice that *knowledge-greedy* and *balanced* are significantly better than *balanced-decreasing* for all φ . As mentioned, a different decay speed d is necessary in this case.

In Figure 7.6 (b), I show different values of p for $\varphi = 0.5$. As before, in terms of Influence the difference between *influence-greedy*, *balanced* and *balanced-decreasing* is not significant, and all algorithms are significantly better than *knowledge-greedy*. In terms of Knowledge, the influence probability does not affect *knowledge-greedy*, *balanced* nor *balanced-decreasing* significantly, as expected. This time, however, *influence-greedy* obtains a significantly better result for $p = 0.1$ than for $p = 0.5$. This may happen because for $p = 0.1$ *influence-greedy* has a higher tendency of selecting nodes with a high number of neighbors, which also tends to be the ones with large teaching lists. Note that this does not happen when the teaching and neighbor lists are independent.

Figure 7.7 shows the regret (lower results are better). We can notice similar results as before: all algorithms have similar regret (except for *knowledge-greedy*), and the regret for *balanced* and *balanced-decreasing* when $\varphi = 0.1$ and $p = 0.5$ is again significantly lower than *influence-greedy* ($\rho = 0.019$ and 0.009 , respectively). This time, however, we can notice that for some parametrizations *balanced-decreasing* is actually the algorithm with the lowest regret. For $p = 0.5$ and $\varphi = 0.5$, *balanced-decreasing* is better than *balanced* with $\rho = 4.7 \times 10^{-4}$. Hence, even though *balanced-decreasing* performed relatively worse than under the uniform model in terms of Knowledge, it is actually the best algorithm in terms of Influence for some parametrizations.

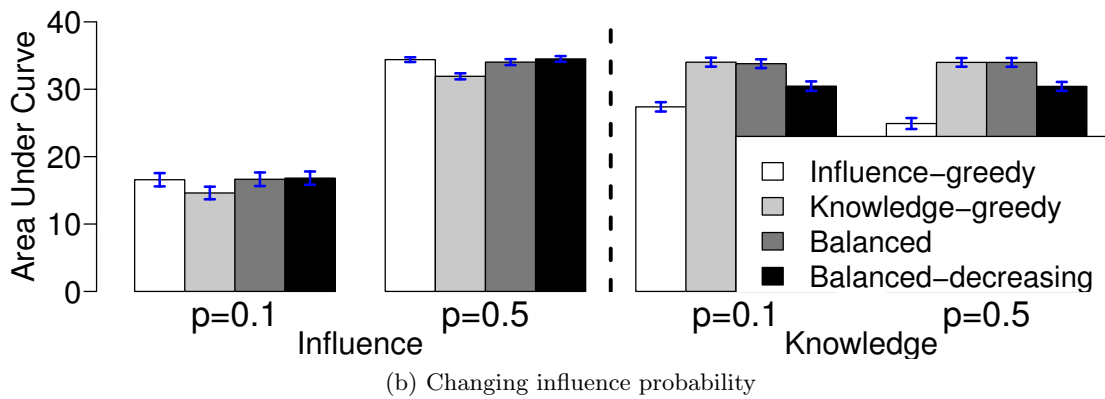
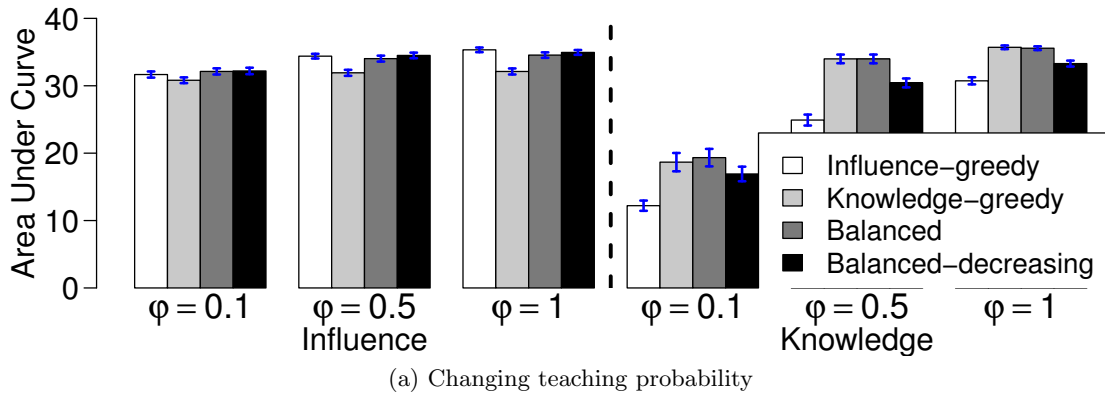


Figure 7.6: Influence and Knowledge for different teaching and influence probabilities (power law distribution).

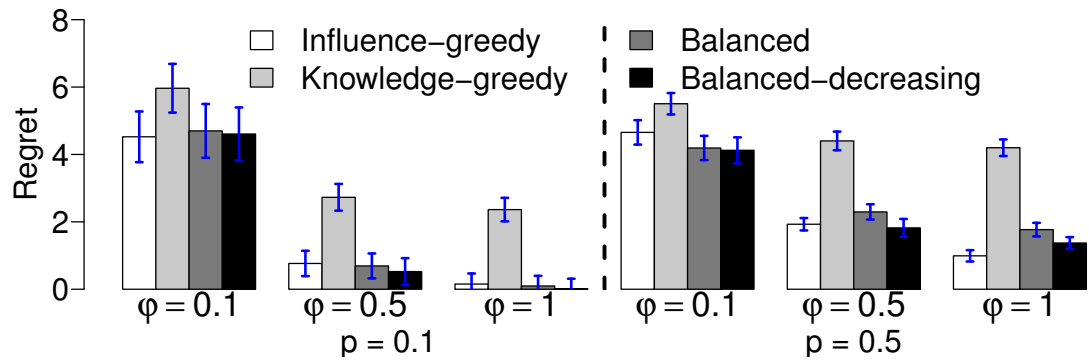


Figure 7.7: Regret for different teaching and influence probabilities (power law distribution). Lower results are better.

7.4 Discussion

I present, for the first time, the problem of simultaneously influencing and mapping a social network. This is a fundamental step for a greater applicability of influencing social networks, for example to increase the adoption of healthy-conscious behavior.

My model, however, currently has two main shortcomings. First, I focus in this chapter on the problem of learning the network nodes, and I assume to always know the edges of the current known subgraph. This assumption was necessary to keep the clarity and precision of my theoretical analysis. A simple extension would be to consider the teaching lists to contain both nodes and edges. That is, at each time we pick a node v_i , we would learn a set of nodes V^i and a set of edges E^i . We must enforce, however, that $v_a, v_b \in V^i$, for all $(v_a, v_b) \in E^i$. Otherwise, we could run into the situation of learning an edge e without knowing the edge end-points, which would be inconsistent. My theoretical results would still apply, but this constraint makes the theoretical analysis unnecessarily more intricate. Another alternative would be to always consider fully connected graphs, but with a very low influence probability. We could, then, increase the influence probability of the edges which we are certain of their existence.

Second, there is not yet a grounded model available to simulate how much someone may be able to teach us about the social network upon being called for interview, nor real world datasets that we could use. I do have datasets of real social network graphs, but as I mentioned, the teaching list of a node is not the same as its neighbor list. People often know others who they do not have direct contact with, or are not “close” enough to have a chance of influencing behavior. Besides, a person may not teach us about all her connections during an interview, due, for example, of shame or privacy issues. An exciting venue for future work would be to perform experiments with human subjects to understand how the teaching lists are actually formed. Such an experiment, however, would present many challenges. To start with, we would need access to the full knowledge of a person, in order to know which persons she fails to report to us. Also, we can expect high variance in the difference between the full knowledge and the teaching lists, as it depends on the personality of each individual.

In my theoretical analysis, I assume a uniform distribution model. The uniform distribution has as its main advantage the lack of any biases, which suits this situation where the precise real-world model is yet unknown. However, intuitively many readers would expect that a person would be more likely to report people who she is directly connected to, and less likely to report people that are more distant in the graph. Hence, I complement my empirical evaluation with experiments where I assume a power law distribution

when generating the teaching lists of each node. My main conclusions still hold under this alternative model, showing that my results generalize to different distributions.

I noticed, however, that although *balanced-decreasing* had a similar performance as *balanced* in terms of Knowledge under the uniform distribution model, it did have a lower performance under the power law model. I used the same decay speed in both cases, but under the power law it is harder to map the network. Hence, in that situation it would be necessary to use a lower decay speed in order to better match the performance of *balanced*. *Balanced-decreasing*, however, is not worse in terms of Influence. In fact, for some parametrizations it was actually the best algorithm, outperforming *influence-greedy*. Therefore, it can perform reasonably well, even without an accurate parametrization of the speed of decay.

Additionally, another interesting point of my experimental analysis is that under both models we found that *influence-greedy* tends to perform as well as *balanced* in terms of Influence. This shows that the traditional greedy influence maximization algorithm is able to perform well even without full knowledge of the social network graph. In fact, for some parametrizations it is even not statistically different than running with full knowledge of the network. There are cases, however, where it will perform badly: for a low teaching probability the regret is lower for the other algorithms, and my Observation 7.2.2 shows that it can have an arbitrarily low performance.

Moreover, *influence-greedy* clearly performs worse in terms of Knowledge. Besides helping to influence a network, mapping the graph is also an important action *per se*. Social agencies usually need knowledge about a community in order to decide their policies and programs, enhancing their ability to increase the general well-being of a population. For instance, during an interview we will not necessarily only learn about the *existence* of a node v_x . When we learn about v_x we may also query for additional information about the person represented by v_x , such as gender, age group or profession. Agencies, then, will be able to use this data when deciding their educational programs.

7.5 Conclusion

I introduced the novel problem of simultaneously influencing and learning the graph of a social network. I show theoretically and experimentally that an algorithm which locally maximizes both influence and knowledge performs as well as an influence-only greedy algorithm in terms of influence, and as well as a knowledge-only greedy approach in terms of knowledge. I present an approximation of my algorithm that gradually decreases the weight given to knowledge-gain, which requires fewer assumptions.

I run experiments using four real-life social networks, where I study two different ways of modeling the knowledge gained by interviewing each node: one where a uniform distribution is assumed, avoiding biases and matching my theoretical analysis; and another where a power law distribution is assumed, and thus nodes are more likely to report others which are closer to them in the social network graph. By testing two different models I validate my conclusions irrespective of the uniform distribution assumption.

My empirical results show not only that the proposed algorithms are competitive with the traditional greedy one in terms of influence, but that they can also significantly influence more nodes than the traditional algorithm when nodes have a low teaching probability. Additionally, the proposed algorithms are significantly better in terms of mapping the graph. Besides helping in influencing, learning about the social network is also important for institutions to effectively decide their policies and educational programs.

Part IV

Team Assessment

Chapter 8

EVERY TEAM DESERVES A SECOND CHANCE

*In the middle of the road there was a stone
there was a stone in the middle of the road
there was a stone
in the middle of the road there was a stone.*

*Never should I forget this event
in the life of my fatigued retinas.
Never should I forget that in the middle of the road
there was a stone
there was a stone in the middle of the road
in the middle of the road there was a stone.
(Carlos Drummond de Andrade)*

8.1 Introduction

It is well known that aggregating the opinions of different agents can lead to a significant performance improvement when solving complex problems. In particular, voting has been extensively used to improve the performance in machine learning [Polikar, 2012], crowdsourcing [Mao et al., 2013, Bachrach et al., 2012], and even board games [Obata et al., 2011, Soejima et al., 2010]. Additionally, it is an aggregation technique that does not depend on any domain, being very suited for wide applicability. However, a team of voting agents will not always be successful in problem-solving. It is fundamental, therefore, to be able to quickly assess the performance of teams, so that a system operator can take actions to recover the situation in time. Moreover, complex problems are generally characterized by a large action space, and hence methods that work well in such situations are of particular interest.

Current works in the multi-agent systems literature focus on identifying faulty or erroneous behavior [Khalastchi et al., 2014, Lindner and Agmon, 2014, Tarapore et al., 2013, Bulling et al., 2013], or verifying correctness of systems [Doan et al., 2014]. Such approaches are able to identify if a system is not operating correctly, but provide no help if a correct system of agents is failing to solve a complex problem. Other works focus on team analysis. Raines et al. [2000] present a method to automatically analyze the performance of a team. The method, however, only works offline and needs domain knowledge. Other methods for team analysis are heavily tailored for robot-soccer [Ramos and Ayanegui, 2008] and focus on identifying opponent tactics [Mirchevska et al., 2014].

In fact, many works in robotics propose monitoring a team by detecting differences in the internal state of the agents (or disagreements), mostly caused by malfunction of the sensors/actuators [Kaminka and Tambe, 1998, Kaminka, 2006, Kalech and Kaminka, 2007, Kalech et al., 2011]. In a system of voting agents, however, disagreements are inherent in the coordination process and do not necessarily mean that an erroneous situation has occurred due to such malfunction. Additionally, research in social choice is mostly focused on studying the guarantees of finding the optimal choice given a noise model for the agents and a voting rule [Caragiannis et al., 2013, List and Goodin, 2001, Conitzer and Sandholm, 2005], but provide no help in assessing the performance of a team of voting agents.

There are also many recent works presenting methods to analyze and/or make predictions about *human* teams playing sports games. Such works use an enormous amount of data to make predictions about many popular sports, such as American football [Quenzel and Shea, 2014, Heiny and Blevins, 2011], soccer [Bialkowski et al., 2014, Lucey et al., 2015] and basketball [Maheswaran et al., 2012, Lucey et al., 2014]. Clearly, however, these works are not applicable to analyzing the performance of a team of voting agents.

Hence, in this chapter, I show a novel method to predict the final performance (success or failure) of a team of voting agents, without using any domain knowledge. Therefore, my method can be easily applied in a great variety of scenarios. Moreover, my approach can be quickly applied online at any step of the problem-solving process, allowing a system operator to identify when the team is failing. This can be useful in many applications. For example, consider a complex problem being solved on a cluster of computers. It is undesirable to allocate more resources than necessary, but if we notice that a team is failing in problem solving, we might wish to increase the allocation of resources. Or consider a team playing together a game against an opponent (such as board games, or poker). Different teams might play better against different opponents. Hence, if we notice that a team is predicted to perform poorly, we could dynamically change it. Under time constraints, however, such prediction must be done quickly.

Although related, note that the contribution of this chapter *is not* in using a team of experts to solve a problem or make predictions [Cesa-Bianchi and Lugosi, 2006] or aggregating multiple classifiers through voting as in ensemble systems [Polikar, 2012]. My objective is, given a team of voting agents, to make a prediction about such a team, in order to estimate whether they will be able to solve a certain problem or not.

My approach is based on a prediction model derived from a graphical representation of the problem-solving process, where the final outcome is modeled as a random variable that is influenced by the subsets of agents that agreed together over the actions taken at each step towards solving the problem. Hence, my representation depends uniquely on the coordination method, and has no dependency on the domain. I explain theoretically why we can make accurate predictions, and I also show the conditions under which we can use a reduced (and scalable) representation. Moreover, my theoretical development allows us to anticipate situations that would not be foreseen by a simple application of classical voting theories. For example, my model indicates that the accuracy can be better for diverse teams composed of different agents than for uniform teams, and that we can make equally accurate predictions for teams that have significant differences in playing strength (which is later confirmed in our experiments). I also study the impact of increasing the action space in the quality of my predictions, and show that we can make better predictions in problems with large action spaces.

I present experimental results in two different domains: Computer Go and Ensemble Learning. In the Computer Go domain, I predict the performance of three different teams of voting agents: a diverse, a uniform, and an intermediate team (with respect to diversity); in four different board sizes. I study the predictions at every turn of the games, and compare with an analysis performed by using an in-depth search. I am able to achieve an accuracy of 71% for a diverse team in 9×9 Go, and of 81% when I increase the action space size to 21×21 Go. For a uniform team, I obtain 62% accuracy in 9×9 , and 75% accuracy in 21×21 Go.

I evaluate different classification thresholds using *Receiver Operating Characteristic* (ROC) curves, and compare the performance for different teams and board sizes according to the area under the curves (AUC). I experimentally show in such analysis that: (i) we can effectively make high-quality predictions for all teams and board sizes; (ii) the quality of my predictions is better for the diverse and intermediate teams than uniform (irrespective of their strength), across all thresholds; (iii) the quality of the predictions increases as the board size grows. Moreover, the impact of increasing the action space on the prediction quality occurs earlier in the game for the diverse team than for the uniform team. Finally, I study the learned prediction functions, and how they change

across different teams and different board sizes. My analysis shows that the functions are not only highly non-trivial, but in fact even open new questions for further study.

In the Ensemble Learning domain, I predict the performance of classifiers that vote to assign labels to set of items. I use the scikit-learn’s digits dataset [Pedregosa et al., 2011], and teams vote to correctly identify hand-written digits. I am also able to obtain high-quality predictions online about the final performance of two different teams of classifiers, showing the applicability of my approach to different domains.

8.2 Prediction Method

I start by presenting my prediction method, and in Section 8.3 I will explain why the method works. I consider scenarios where agents vote at every step (i.e., world state) of a complex problem, in order to take common decisions at every step towards problem-solving. Formally, let \mathbf{T} be a set of agents t_i , \mathbf{A} be a set of actions a_j and \mathbf{S} be a set of world states s_k . The agents must vote for an action at each world state, and the team takes the action decided by the *plurality voting rule*, that picks the action that received the highest number of votes (I assume ties are broken randomly). The team obtains a final reward r upon completing all world states. In this chapter, I assume two possible final rewards: “success” (1) or “failure” (0).

I define the prediction problem as follows: without using any knowledge of the domain, identify the final reward that will be received by a team. This prediction must be executable at any world state, allowing a system operator to take remedial procedures in time.

I now explain my algorithm. The main idea is to learn a prediction function, given the frequencies of agreements of all possible agent subsets over the chosen actions. In order to learn such function, we need to define a feature vector to represent each problem solving instance (for example, a game). My feature vector records the frequency that each subset of agents was the one that determined the action taken by the team (i.e., the subset whose action was selected as the action of the team by the plurality voting rule). When learning the prediction function I calculate the feature vector considering the whole history of the problem solving process (for example, from the first to the last turn of a game). When using the learned prediction function to actually perform a prediction, we can simply compute the feature vector considering all history from the first world state to the current one (for example, from the first turn of a game up to the current turn), and use it as input to our learned function. Note that my feature vector does not hold any domain information, and uses solely the *voting patterns* to represent the problem solving instances.

Formally, let $\mathcal{P}(\mathbf{T}) = \{\mathbf{T}_1, \mathbf{T}_2, \dots\}$ be the power set of the set of agents, a_i be the action chosen in world state s_j and $\mathbf{H}_j \subseteq \mathbf{T}$ be the subset of agents that agreed on a_i in that world state. Consider the feature vector $\vec{x} = (x_1, x_2, \dots)$ computed at world state s_j , where each dimension (feature) has a one-to-one mapping with $\mathcal{P}(\mathbf{T})$. I define x_i as the *proportion* of times that the chosen action was agreed upon by the subset of agents \mathbf{T}_i . That is,

$$x_i = \sum_{k=1}^{|\mathbf{S}_j|} \frac{\mathbb{I}(\mathbf{H}_k = \mathbf{T}_i)}{|\mathbf{S}_j|},$$

where \mathbb{I} is the indicator function and $\mathbf{S}_j \subseteq \mathbf{S}$ is the set of world states from s_1 to the current world state s_j .

Hence, given a set $\tilde{\mathbf{X}}$ such that for each feature vector $\vec{x}_t \in \tilde{\mathbf{X}}$ we have the associated reward r_t , we can estimate a function, \hat{f} , that returns an estimated reward between 0 and 1 given an input \vec{x} . I classify estimated rewards above a certain threshold ϑ (for example, 0.5) as “success”, and below it as “failure”.

In order to *learn* the classification model, the features are computed at the final world state. That is, the feature vector will record the frequency that each possible subset of agents won the vote, calculated from the first world state to the last one. For each feature vector, we have (for learning) the associated reward: “success” (1) or “failure” (0), accordingly with the final outcome of the problem solving process. In order to *execute* the prediction, the features are computed at the current world state (i.e., all history of the current problem solving process from the first world state to the current one).

I use classification by logistic regression, which models \hat{f} as

$$\hat{f}(\vec{x}) = \frac{1}{1 + e^{-(\alpha + \vec{\beta}^T \vec{x})}},$$

where α and $\vec{\beta}$ are parameters that will be learned given $\tilde{\mathbf{X}}$ and the associated rewards. While training, I eliminate two of the features. The feature corresponding to the subset \emptyset is dropped because an action is chosen only if at least one of the agents voted for it. Also, since the rest of the features sum up to 1, and are hence linearly dependent, I also drop the feature corresponding to all agents agreeing on the chosen action.

I also study a variant of this prediction method, where we use only information about the number of agents that agreed upon the chosen action, but not which agents exactly were involved in the agreement. For that variant, I consider a reduced feature vector

	Agent 1	Agent 2	Agent 3
Iteration 1	a_1	a_1	a_2
Iteration 2	a_2	a_2	a_1
Iteration 3	a_1	a_2	a_2

Table 8.1: A simple example of voting profiles after three iterations of problem-solving.

$\vec{y} = (y_1, y_2, \dots)$, where I define y_i to be the proportion of times that the chosen action was agreed upon by any subset of i agents:

$$y_i = \sum_{k=1}^{|\mathbf{S}_j|} \frac{\mathbb{I}(|\mathbf{H}_k| = i)}{|\mathbf{S}_j|},$$

where \mathbb{I} is the indicator function and $\mathbf{S}_j \subseteq \mathbf{S}$ is the set of world states from s_1 to the current world state s_j . I compare the two approaches in Section 8.4.

8.2.1 Example of Features

I give a simple example to illustrate my proposed feature vectors. Consider a team of three agents: t_1, t_2, t_3 . Let's assume three possible actions: a_1, a_2, a_3 . Consider that, in three iterations of the problem solving process, the voting profiles were as shown in Table 8.1, where I show which action each agent voted for at each iteration. Based on plurality voting rule, the action chosen for the respective iterations would be a_1, a_2 , and a_2 .

We can see an example of how the full feature vector will be defined at each iteration in Table 8.2, where each column represents a possible subset of the set of agents, and I mark the frequency that each subset agreed on the chosen action. Note that the frequency of the subsets $\{t_1\}$, $\{t_2\}$ and $\{t_3\}$ remains 0 in this example. This happens because we only count the subset where all agents involved in the agreement are present. If there was a situation where, for example, agent t_1 votes for a_1 , agent t_2 votes for a_2 and agent t_3 votes for a_3 , then we would select one of these agents by random tie breaking. After that, we would increase the frequency of the corresponding subset containing only the agent that was chosen (i.e, either $\{t_1\}$, $\{t_2\}$ or $\{t_3\}$).

If the problem has only three iterations in total, we would use the feature vector at the last iteration and the corresponding result (“success” — 1, or “failure” — 0), while learning the function \hat{f} (that is, the feature vectors at Iteration 1 and 2 would be ignored). If, however, we already learned a function \hat{f} , then we could use the feature vector at Iteration 1, 2 or 3 as input to \hat{f} to execute a prediction. Note that at Iteration 1 and 2 the output (i.e., the prediction) of \hat{f} will be exactly the same, as the feature vector did not change. At Iteration 3, however, we may have a different output/prediction.

	$\{t_1\}$	$\{t_2\}$	$\{t_3\}$	$\{t_1, t_2\}$	$\{t_1, t_3\}$	$\{t_2, t_3\}$
Iteration 1	0	0	0	1	0	0
Iteration 2	0	0	0	1	0	0
Iteration 3	0	0	0	2/3	0	1/3

Table 8.2: Example of the full feature vector after three iterations of problem solving.

	1	2
Iteration 1	0	1
Iteration 2	0	1
Iteration 3	0	1

Table 8.3: Example of the reduced feature vector after three iterations of problem solving.

In Table 8.3, I show an example of the reduced feature vector, where the column headings define the number of agents involved in an agreement over the chosen action. I consider here the same voting profiles as before, shown in Table 8.1. Note that the reduced representation is much more compact, but we have no way to represent the change in which specific agents were involved in the agreement, from Iteration 2 to Iteration 3. In this case, therefore, we would always have the same prediction after Iteration 1, Iteration 2 and also Iteration 3.

8.3 Theory

I consider here the view of social choice as a way to estimate a “truth”, or the correct (i.e., best) action to perform in a given world state. Hence, we can model each agent as a probability distribution function (pdf): that is, given the correct outcome, each agent will have a certain probability of voting for the best action, and a certain probability of voting for some incorrect action. These pdfs are not necessarily the same across different world states (Chapter 3). Hence, given the voting profile in a certain world state, there will be a probability p of picking the correct choice (for example, by the plurality voting rule).

I will start by developing, in Section 8.3.1, a simple explanation of why we can use the voting patterns to predict success or failure of a team of voting agents, based on classical voting theories. Such explanation will give an intuitive idea of why we can use the voting patterns to predict success or failure of a team of voting agents, and it can be immediately derived from the classical voting models. However, it fails to explain some of the results in Section 8.4, and it needs the assumption that plurality is an optimal voting rule. Hence, it is not enough for a deeper understanding of my prediction methodology. Therefore, I will later present, in Section 8.3.2, my main theoretical model, that provides a better

understanding of my results. In particular, based on classical models we would expect to make better predictions for teams that have a greater performance (i.e., likelihood of being correct). My theory and experiments will show, however, that we can actually make better predictions for teams that are more diverse, even if they have a worse performance. Moreover, I will also be able to build on my previous *ST* agent model, in Section 8.3.3, to study the effect of increasing the action space on the prediction quality.

8.3.1 Classical Voting Model

I start with a simple example to show that we can use the outcome of plurality voting to predict success. Consider a scenario with two agents and two possible actions, a correct and an incorrect one. I assume, for this example, that agents have a probability of 0.6 of voting for the correct action and 0.4 of making a mistake.

If both agents vote for the same action, they are either both correct or both wrong. Hence, the probability of the team being correct is given by $0.6^2/(0.6^2 + 0.4^2) = 0.69$. Therefore, if the agents agree, the team is more likely correct than wrong. If they vote for different actions, however, one will be correct and the other one wrong. Given that profile, and assuming that we break ties randomly, the team will have a 0.5 probability of being correct. Hence, the team has a higher probability of taking a correct choice when the agents agree than when they disagree ($0.69 > 0.5$). Therefore, if across multiple iterations these agents agree often, the team has a higher probability of being correct across these iterations, and we can predict that the team is going to be successful. If they disagree often, then the probability of being correct across the iterations is lower, and we can predict that the team will not be successful.

More generally than the previous example, we can consider all cases where plurality is the optimal voting rule. In social choice, optimal voting rules are often studied as maximum likelihood estimators (MLE) of the correct choice [Conitzer and Sandholm, 2005]. That is, each agent is modeled as having a noisy perception of the truth (or correct outcome). Hence, the correct outcome influences how each agent is going to vote, as shown in the model in Figure 8.1. For example, consider a certain agent t that has a probability 0.6 of voting for the best action, and let's say we are in a certain situation where action a^* is the best action. In this situation agent t will have a probability of 0.6 of voting for a^* .

Therefore, given a voting profile and a noise model (the probability of voting for each action, given the correct outcome) of each agent, we can estimate the likelihood of each action being the best by a simple (albeit computationally expensive) probabilistic inference. Any voting rule is going to be optimal if it corresponds to always picking the action that has the maximum likelihood of being correct (i.e., the action with maximum

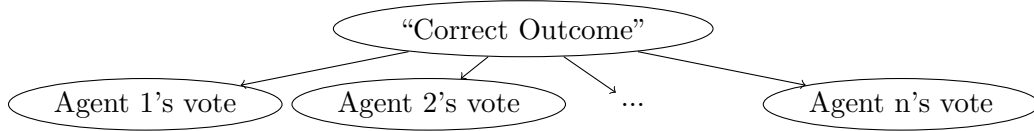


Figure 8.1: “Classical” voting model. Each agent has a noisy perception of the truth, or correct outcome. Hence, its vote is influenced by the correct outcome.

likelihood of being the best action), according to the assumed noise model of the agents. That is, the output of an optimal voting rule always corresponds to the output of actually computing, by the probabilistic inference method mentioned above, which action has the highest likelihood of being the best one.

If plurality is assumed to be an optimal voting rule, then the action voted by the largest number of agents has the highest probability of being the optimal action. We can expect, therefore, that the higher the number of agents that votes for an action, the higher the probability that the action is the optimal one. Hence, given two different voting profiles with a different number of agreeing agents, we expect that the team has a higher probability of being correct (and, therefore, be successful) in the voting profile where a larger number of agents agrees on the chosen action.

I formalize this idea in the following proposition, under the classical assumptions of voting models. Hence, I consider that the agents have a higher probability of voting for the best action than any other action (which makes plurality a MLE voting rule [List and Goodin, 2001]), uniform priors over all actions, and that all agents are identical and independent.

Proposition 8.3.1 *The probability that a team is correct increases with the number of agreeing agents m in a voting profile, if plurality is MLE.*

Proof: Let a^* be the best action (whose identity we do not know) and $\mathbf{V} = v_1, v_2 \dots v_n$ be the votes of n agents. The probability of any action a being the best action, given the votes of the agents (i.e., $P(a = a^* | v_1, v_2 \dots v_n)$), is governed by the following relation:

$$P(a = a^* | v_1, v_2 \dots v_n) \propto P(v_1, v_2 \dots v_n | a = a^*)P(a = a^*) \quad (8.1)$$

Let’s consider two voting profiles $\mathbf{V}_1, \mathbf{V}_2$, where in one a higher number of agents agree in the chosen action than in the other (i.e., $m_{\mathbf{V}_1} > m_{\mathbf{V}_2}$). Let w_1 be the action with the highest number of votes in \mathbf{V}_1 , and w_2 the one in \mathbf{V}_2 .

Without loss of generality (since the order does not matter), let’s reorder the voting profiles \mathbf{V}_1 and \mathbf{V}_2 , such that all votes for w_1 are in the beginning of \mathbf{V}_1 and all votes

for w_2 are in the beginning of \mathbf{V}_2 . Now, let \mathbf{V}_1^x and \mathbf{V}_2^x be the voting profiles considering only the first x agents (after reordering).

We have that $P(\mathbf{V}_1^{m_{\mathbf{V}_2}}|w_1 = a^*) = P(\mathbf{V}_2^{m_{\mathbf{V}_2}}|w_2 = a^*)$, since up to the first $m_{\mathbf{V}_2}$ agents for both voting profiles we are considering the case where all agents voted for a^* .

Now, let's consider the agents from $m_{\mathbf{V}_2} + 1$ to $m_{\mathbf{V}_1}$. In \mathbf{V}_1 , the voted action of all agents (still w_1) is wired to a^* (by the conditional probability). However, in \mathbf{V}_2 , the voted action $a \neq w_2$ is not wired to a^* in the conditional probability anymore. As each agent is more likely to vote for a^* than any other action, from $m_{\mathbf{V}_2} + 1$ to $m_{\mathbf{V}_1}$, the events in \mathbf{V}_1 (an agent voting for a^*) has higher probability than the events in \mathbf{V}_2 (an agent voting for an action $a \neq a^*$). Hence, $P(\mathbf{V}_1^{m_{\mathbf{V}_1}}|w_1 = a^*) > P(\mathbf{V}_2^{m_{\mathbf{V}_1}}|w_2 = a^*)$.

Now let's consider the votes after $m_{\mathbf{V}_1}$. In \mathbf{V}_1 there are no more votes for w_1 , and in \mathbf{V}_2 there are no more votes for w_2 . Hence, all the subsequent votes are not wired to any ranking (as we only wire $w_1 = a^*$ and $w_2 = a^*$ in the conditional probabilities). Therefore, each vote can be assigned to any ranking position that is not the first. Since the agents are independent, any sequence of votes will thus be as likely. Hence, $P(\mathbf{V}_1|w_1 = a^*) > P(\mathbf{V}_2|w_2 = a^*)$.

Since I assume uniform priors, it follows that:

$$P(w_1 = a^*|\mathbf{V}_1) > P(w_2 = a^*|\mathbf{V}_2)$$

Therefore, the team is more likely correct in profiles where a higher number of agents agree. ■

Hence, if across multiple voting iterations, a higher number of agents agree often, we can predict that the team is going to be successful. If they disagree a lot, we can expect that they are wrong in most of the voting iterations, and we can predict that the team is going to fail.

In the next observation I show that we can increase the prediction accuracy by knowing not only how many agents agreed, but also which specific agents were involved in the agreement. Basically, I show that the probability of a team being correct depends on the agents involved in the agreement. Therefore, if we know that the best agents are involved in an agreement, we can be more certain of a team's success. This observation motivates the use of the full feature vector, instead of the reduced one.

Observation 8.3.2 *Given two profiles $\mathbf{V}_1, \mathbf{V}_2$ with the same number of agreeing agents m , the probability that a team is correct is not necessarily equal for the two profiles.*

We can easily show by an example (that is, we only need one example where the probability is not equal to show that it will not always be equal). Consider a problem

with two actions. Consider a team of three agents, where t_1 and t_2 have a probability of 0.8 of being correct, while t_3 has a probability of 0.6 of being correct. As the probability of picking the correct action is the highest for all agents, the action chosen by the majority of the agents has the highest probability of being correct (that is, we are still covering a case where plurality is MLE).

However, when only t_1 and t_2 agree, the probability that the team is correct is given by: $0.8^2 \times 0.4 / (0.8^2 \times 0.4 + 0.2^2 \times 0.6) = 0.91$. When only t_2 and t_3 agree, the probability that the team is correct is given by: $0.8 \times 0.6 \times 0.2 / (0.8 \times 0.6 \times 0.2 + 0.2 \times 0.4 \times 0.8) = 0.59$. Hence, the probability that the team is correct is higher when t_1 and t_2 agree than when t_2 and t_3 agree.

However, based solely on the classical voting models, one would expect that given two different teams, the predictions would be more accurate for the one that has greater performance (i.e., likelihood of being correct), as I formalize in the following proposition. Therefore, this model fails to explain my experimental results (as I will show later). I will use the term *strength* to refer to a team's performance.

Proposition 8.3.3 *Under the classical voting models, given two different teams, one can expect to make better predictions for the strongest one.*

Proof Sketch: Under the classical voting models, assuming the agents have a noise model such that plurality is a MLE, we have that the best team will have a greater probability of being correct given a voting profile where m agents agree than a worse team with the same amount of m agreeing agents.

Hence, the probability of the best team being correct will be closer to 1 in comparison with the probability of the worse team being correct. The closer the probability of success is to 1, the easier it is to make predictions. Consider a Bernoulli trial with probability of success $p \approx 1$. In the learning phase, we will see many successes accordingly. In the testing phase, we will predict the majority of the two for every trial, and we will go wrong only with probability $|1 - p| \approx 0$.

Of course, we could also have an extremely weak team, that is wrong most of the time. For such team, it would also be easy to predict that the probability of success is close to 0. Notice, however, that I am assuming here the classical voting models, where plurality is a MLE. In such models, the agents must play "reasonably well": classically they are assumed to have either a probability of being correct greater than 0.5 or the probability of voting for the best action is the highest one in their pdf [List and Goodin, 2001]. Otherwise, plurality is not going to be a MLE. ■

Consider, however, that the strongest team is composed of copies of the best agent (which would often be the case, under the classical assumptions). We actually have

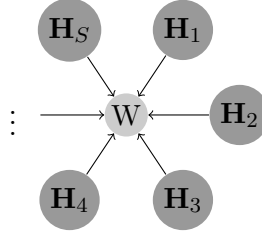


Figure 8.2: *My main model.* I assume that the subset of agents that decided the action of the team at each world state \mathbf{H}_i determines whether the team will be successful or not (W).

that, in fact, such agents will not necessarily have noise models (pdfs) where the best action has the highest probability in all world states. In some world states, a suboptimal action could have the highest probability, making the agents agree on the same mistakes (Chapter 3). Therefore, when plurality is not actually a MLE in all world states, we have that Proposition 8.3.1 will not hold in the world states where this happens. Hence, we will predict that the team made a correct choice, when actually the team was wrong, causing problems in our accuracy. I give more details in the next section.

8.3.2 Main Theoretical Model

I now present my main theory, that holds irrespective of plurality being an optimal voting rule (MLE) or not. Again, I consider agents voting across multiple world states. I assume that all iterations equally influence the final outcome, and that they are all independent.

Let the final reward of the team be defined by a random variable W , and let the number of world states be S . I model the problem solving process by the graphical model in Figure 8.2, where \mathbf{H}_j represents the subset of agents that agreed on the chosen action at world state s_j . That is, I assume that the subset of agents that decided the action taken by the team at each world state of the problem solving process will determine whether the team will be successful or not in the end. A specific problem (for example, Go games where the next state will depend on the action taken in the current one) would call for more complex models to be completely represented. My model is a simplification of the problem solving process, abstracting away the details of specific problems.

For any subset \mathbf{H} , let $P(\mathbf{H})$ be the probability that the chosen action was correct given the subset of agreeing agents. If the correct action is a^* , $P(\mathbf{H})$ is equivalent to:

$$\frac{P(\forall t \in \mathbf{H}, t \text{ chooses } a^*, \quad \forall t \notin \mathbf{H}, t \text{ chooses } a \neq a^*)}{P(\exists a' \quad \forall t \in \mathbf{H}, t \text{ chooses } a', \quad \forall t \notin \mathbf{H}, t \text{ chooses } a \neq a')}$$

where \mathbf{H} is the subset of agents which voted for the action taken by the team.

Note that $P(\mathbf{H})$ depends on both the team and the world state. However, I marginalize the probabilities to produce a value that is an average over all world states. I consider that, for a team to be successful, there is a threshold δ such that:

$$\left\{ \prod_{j=1}^S P(\mathbf{H}_j) \right\}^{1/S} > \delta \quad (8.2)$$

I use the exponent $1/S$ in order to maintain a uniform scale across all problems. Each problem might have a different number of world states; and for one with many world states, it is likely that the incurred product of probabilities is sufficiently low to fail the above test, independent of the actual subsets of agents that agreed upon the chosen actions. However, the final reward is not dependent on the number of world states.

I can show, then, that we can use a linear classification model (such as logistic regression) that is equivalent to Equation 8.2, to predict the final reward of a team.

Theorem 8.3.4 *Given the model in Equation 8.2, the final outcome of a team can be predicted by a linear model over agreement frequencies.*

Proof: Getting the log in both sides of Equation 8.2, we have:

$$\sum_{j=1}^S \frac{1}{S} \log(P(\mathbf{H}_j)) > \log(\delta)$$

The sum over the steps (world states) of the problem-solving process can be transformed to a sum over all possible subset of agents that can be encountered, \mathcal{P} :

$$\sum_{\mathbf{H} \in \mathcal{P}} \frac{n_{\mathbf{H}}}{S} \log(P(\mathbf{H})) > \log(\delta), \quad (8.3)$$

where $n_{\mathbf{H}}$ is the number of times the subset of agreeing agents \mathbf{H} was encountered during problem solving. Hence, $\frac{n_{\mathbf{H}}}{S}$ is the frequency of seeing the subset \mathbf{H} , which I will denote by $f_{\mathbf{H}}$.

Recall that \mathbf{T} is the set of all agents. Hence, $f_{\mathbf{T}}$ (which is the frequency of all agents agreeing on the same action), is equal to $1 - \sum_{\mathbf{H} \in \mathcal{P} \setminus \{\mathbf{T}\}} f_{\mathbf{H}}$. Also, note that $n_{\emptyset} = 0$, since at least one agent must pick the chosen action. Equation 8.3 can, hence, be rewritten as:

$$f_{\mathbf{T}} \log(P(\mathbf{T})) + \sum_{\mathbf{H} \in \mathcal{P} \setminus \mathbf{T}} f_{\mathbf{H}} \log(P(\mathbf{H})) > \log(\delta)$$

$$\left(1 - \sum_{\mathbf{H} \in \mathcal{P} \setminus \{\mathbf{T}\}} f_{\mathbf{H}} \right) \log(P(\mathbf{T})) + \sum_{\mathbf{H} \in \mathcal{P} \setminus \mathbf{T}} f_{\mathbf{H}} \log(P(\mathbf{H})) > \log(\delta)$$

$$\log(P(\mathbf{T})) + \sum_{\mathbf{H} \in \mathcal{P} \setminus \mathbf{T}} f_{\mathbf{H}} \log \left(\frac{P(\mathbf{H})}{P(\mathbf{T})} \right) > \log(\delta)$$

Hence, my final model will be:

$$\sum_{\mathbf{H} \in \mathcal{P} \setminus \mathbf{T}} \log \left(\frac{P(\mathbf{H})}{P(\mathbf{T})} \right) f_{\mathbf{H}} > \log \left(\frac{\delta}{P(\mathbf{T})} \right) \quad (8.4)$$

Note that $\log(\frac{\delta}{P(\mathbf{T})})$ and the “coefficients” $\log(\frac{P(\mathbf{H})}{P(\mathbf{T})})$ are all *constants* with respect to a given team, as I have discussed earlier. Considering the set of all $f_{\mathbf{H}}$ (for each possible subset of agreeing agents \mathbf{H}) to be the characteristic features of a single problem, the coefficients can now be *learned* from training data that contains many problems represented using these features. Further, the outcome of a team can be estimated through a linear model. ■

The number of constants is exponential, however, as the size of the team grows. Therefore, in the following corollary, I show that (under some conditions) we can approximate well the prediction with a reduced feature vector that grows linearly. In order to differentiate different possible subsets, I will denote by \mathbf{H}^i a certain subset $\in \mathcal{P}$, and by $|\mathbf{H}^i|$ the size of that subset (i.e., the number of agents that agree on the chosen action).

Corollary 8.3.5 *If $P(\mathbf{H}^i) \approx P(\mathbf{H}^j) \forall \mathbf{H}^i, \mathbf{H}^j$ such that $|\mathbf{H}^i| = |\mathbf{H}^j|$, we can approximate the prediction with a reduced feature vector, that grows linearly with the number of agents. Furthermore, in a uniform team the reduced representation is equal to the full representation.*

Proof: By the assumption of the corollary, there is a $P_{\mathbf{H}^n}$, defined as $P_{\mathbf{H}^n} \approx P(\mathbf{H}^j)$, $\forall \mathbf{H}^j$ such that $|\mathbf{H}^j| = n$. Let $f_n = \sum f_{\mathbf{H}^j}$, over all $|\mathbf{H}^j| = n$. Also, let \mathbf{N}' be the set of all integers $0 < x < N$, where N is the number of agents. We thus have that:

$$\sum_{\mathbf{H} \in \mathcal{P} \setminus \mathbf{T}} f_{\mathbf{H}} \log \left(\frac{P(\mathbf{H})}{P(\mathbf{T})} \right) \approx \sum_{x \in \mathbf{N}'} f_x \log \left(\frac{P_{\mathbf{H}^n}}{P(\mathbf{T})} \right) \quad (8.5)$$

As $P_{\mathbf{H}^n}$ depends only on the number of agents, we have that such representation grows linearly with the size of the team.

Moreover, note that for a team made of copies of the same agent, we have that $P_{\mathbf{H}^n} = P(\mathbf{H}^j)$, $\forall \mathbf{H}^j$ such that $|\mathbf{H}^j| = n$. Hence, the left hand side of Equation 8.5 is going to be equal to the right hand side. ■

Also, notice that my model does not need any assumptions about plurality being an optimal voting rule (MLE). In fact, there are no assumptions about the voting rule at

all. Hence, Proposition 8.3.1, Observation 8.3.2 and Proposition 8.3.3 do not apply, and we can still make accurate predictions irrespective of the performance of a team.

We can also note that the accuracy of the predictions is not going to be the same across different teams. Given two teams where plurality is the optimal voting rule in general (i.e., $P(\mathbf{H}^i) > P(\mathbf{H}^j)$, $\forall \mathbf{H}^i, \mathbf{H}^j$ where $|\mathbf{H}^i| > |\mathbf{H}^j|$): one *uniform*, made of copies of the best agent, and a *diverse*, made of different agents, we have that we can actually make better predictions for *diverse* than *uniform*, irrespective of the actual playing performance of these two teams.

Corollary 8.3.6 *Given a diverse and a uniform team, the accuracy of the prediction for the diverse team can be higher, even if the diverse team has a lower probability of victory.*

Proof Sketch: Note that the *learned* constants $\hat{c}_{\mathbf{H}} \approx \log\left(\frac{P(\mathbf{H})}{P(\mathbf{T})}\right)$ represent the marginal probabilities across all world states. However, I showed (Chapter 3) that the agent with the highest marginal probability of voting for the correct choice will not necessarily have the highest probability of being correct at all world states.

Hence, let **Bad** be the set of world states where the best agent does not have the highest probability of voting for the correct action in its pdf. In such world states, the agents tend to agree over the incorrect action that has the highest probability. This follows from modeling the voting of all agents as a multinomial distribution. Hence, the expected number of agents that vote for an action a_j , in a team with n agents, is given by: $E[|\mathbf{H}|] = n \times p_j$, where p_j is the probability of the agent voting for action a_j . Therefore, the action with the highest probability will tend to receive the highest number of votes.

However, in my prediction model, the estimation that the team is correct will get higher as more agents agree together upon the final choice. Hence, we will tend to make wrong predictions for the world states in the set **Bad**, and our accuracy will be lower as $|\mathbf{Bad}|$ gets higher.

Since a diverse team is composed of agents with different pdfs, it is less likely that in a given world state they will all have the highest probability in the same incorrect action (Chapter 3). Hence, it is less likely that the situation described above happens, and we can have better predictions for a diverse team. ■

Although I only give here a proof sketch, in Section 8.4 I experimentally show a statistically significant higher accuracy for the predictions for a *diverse* team than for a *uniform* team, even though they have similar strength (i.e., performance in terms of winning rates). I am able to achieve a better prediction for *diverse* teams both in the end of the problem-solving process and also while doing online predictions at any world state.

8.3.3 Action Space Size

I present now my study concerning the quality of the predictions over large action space sizes. In order to perform this analysis, I assume the *spreading tail* (*ST*) agent model, presented in Chapter 4. The *ST* agent model was developed to study how teams of voting agents change in performance as the size of the action space increases. The basic assumption is that the pdf of each member of the team has a non-zero probability over an increasingly larger number of suboptimal actions as the action space grows, while the probability of voting for the optimal action remains unchanged. Chapter 4 perform an experimental validation of this model in the Computer Go domain.

As a reminder, I briefly summarize here the formal definition of the *ST* agent model, and I refer the reader to Chapter 4 for a more detailed description. Let \mathbf{D}_m be the set of suboptimal actions ($a_j, j \neq 0$) assigned with a nonzero probability in the pdf of an agent i , and $d_m = |\mathbf{D}_m|$. The *ST* model assumes that there is a bound in the ratio of the suboptimal action with highest probability and the one with lowest nonzero probability, i.e., let $p_{i,min} = \min_{j \in \mathbf{D}_m} p_{i,j}$ and $p_{i,max} = \max_{j \in \mathbf{D}_m} p_{i,j}$; there is a constant ζ such that $p_{i,max} \leq \zeta p_{i,min} \forall$ agents i . *ST* agents are agents whose d_m is non-decreasing on m and $d_m \rightarrow \infty$ as $m \rightarrow \infty$. I consider that there is a constant $\epsilon > 0$, such that for all *ST* agents i , $\forall m, p_{i,0} \geq \epsilon$. I also assume that $p_{i,0}$ does not change with m .

Let the size of the action space $|\mathbf{A}| = \varrho$, and $p_{i,j}$ be the probability that agent i votes for action with rank j . In Chapter 4, I show that when $\varrho \rightarrow \infty$, the probability that a team of n *ST* agents will play the optimal action converges to:

$$\tilde{p}_{best} = 1 - \prod_{i=1}^n (1 - p_{i,0}) - \sum_{i=1}^n (p_{i,0} \prod_{j=1, j \neq i}^n (1 - p_{j,0})) \frac{n-1}{n}, \quad (8.6)$$

that is, the probability of two or more agents agreeing over suboptimal actions converges to zero, and the agents can only agree over the optimal choice (note that a suboptimal action can still be taken as we may have situations where no agent agrees). Hence, Equation 4.1 calculates the total probability minus the cases where the best action is not chosen: the second term covers the case where all agents vote for a suboptimal action and the third term covers the case where one agent votes for the optimal action and all other agents vote for suboptimal actions.

Before proceeding to my study, I am going to make a few definitions and then two weak assumptions. I consider now here any action space size. Let α be the probability of a team taking the optimal action when all agents disagree. Since we can only take the optimal action if one agent votes for that action, α is a function of the probability of each agent voting for the optimal action. That is, we may have voting profiles where all agents disagree and no agent voted for the optimal action, or we may have voting profiles

where all agents disagree, but there is one agent that voted for the optimal action (and, hence, we may still take the optimal action due to random tie breaking).

Let β be the probability of a team taking the optimal action when there is some agreement on the voting profile. β may be different according to each voting profile, but we assume that we always have that $\beta < 1$ if $\varrho < \infty$, and $\beta = 1$ if $\varrho \rightarrow \infty$, according to the *ST* agent model. That is, if two or more agents agree, there is always some probability $q > 0$ that they are agreeing over a suboptimal action, and $q \rightarrow 0$ as $\varrho \rightarrow \infty$.

I will make the following weak assumptions: (i) If there is no agreement, the team is more likely to take a suboptimal action than an optimal action. I.e., $\alpha < 1 - \alpha$; (ii) If there is agreement, there is at least one voting profile where the team is more likely to take an optimal action than a suboptimal action. That is, there is at least one β such that $\beta > 1 - \beta$.

Assumption (i) is weak, since $\alpha < 1/n$ (as we break ties randomly and there may be cases where no agent votes for the optimal action). Clearly $1/n < 1 - 1/n$ for $n > 2$. Assumption (ii) is also weak, because if we are given a team that is always more likely to take suboptimal actions than an optimal action for *any* voting profile, then a trivial predictor that always outputs “failure” would be optimal (and, hence, we would not need a prediction at all). Therefore, assumption (i) and (ii) are satisfied for all situations of interest. I present now my result:

Theorem 8.3.7 *Let \mathbf{T} be a set of ST agents. The quality of our prediction about the performance of \mathbf{T} is the highest as $\varrho \rightarrow \infty$.*

Proof: Let’s fix the problem to predicting performance at one world state. Hence, as we consider a single decision, there is a single \mathbf{H}^i such that $f_{\mathbf{H}^i} = 1$, and $f_{\mathbf{H}^j} = 0 \forall j \neq i$. In order to simplify the notation, I denote by \mathcal{H} the subset \mathbf{H}^i corresponding to $f_{\mathbf{H}^i} = 1$. I also consider the performance of the team as “success” on that fixed world state if they take the optimal action, and as “failure” otherwise.

Let a *voting event* be the process of querying the agents for the vote, obtaining the voting profile and the corresponding final decision. Hence, it has a unique correct label (“success” or “failure”). A voting event ξ will be mapped to a point χ in the feature space, according to the subset of agents that agreed on the chosen action. Multiple voting events, however, will be mapped to the same point χ (as exactly the same subset can agree in different situations, sometimes they may be agreeing over the optimal action, and sometimes they may be agreeing over suboptimal actions). Hence, given a point χ , there is a certain probability that the team was successful, and a certain probability that the team failed. Therefore, by assigning a label to that point, our predictor will also be correct with a certain probability. With enough data, the predictor will output the

more likely of the two events. That is, if given a profile, the team has a probability p of taking the optimal action, and $p > 1 - p$, the predictor will output “success”, and it will be correct with probability p . Correspondingly, if $1 - p > p$, the predictor will output “failure”, and it will be correct with probability $1 - p$. Hence, the probability of the prediction being correct will be $\max(p, 1 - p)$.

I first study the probability of making a correct prediction across the whole feature space, for different action space sizes, and after that I will focus on what happens with the specific voting events as the action space changes.

Let us start by considering the case when $\varrho \rightarrow \infty$. By Equation 4.1, we know that every time two or more agents agree on the same action, that action will be the optimal one. Note that this is a very clear division of the feature space, as for every single point where $|\mathcal{H}| \geq 2$ the team will be successful with probability 1. Therefore, on this subspace we can make perfect predictions.

The only points in the feature space where a team may still take a suboptimal action are the ones where a single agent agrees on the chosen action, i.e., $|\mathcal{H}| = 1$. Hence, for such points we will make a correct prediction with probability $\max(\alpha, 1 - \alpha)$.

Let’s now consider cases with a smaller action space size (i.e., $\varrho < \infty$). Let’s first consider the subspace $|\mathcal{H}| \geq 2$. Before, our predictor was correct with probability 1. Now, given a voting event where there is an agreement, there will be a probability $\beta < 1$ of the team taking the optimal action. Hence, the predictor will be correct with probability $\max(\beta, 1 - \beta)$, but $\max(\beta, 1 - \beta) < 1$.

Let’s consider now the subspace $|\mathcal{H}| = 1$. Here the quality of the prediction depends on α , which is a function of the probability of each agent playing the best action. On the *ST* agent model, however, the probability of one agent voting for the best action is independent of ϱ (Chapter 4). Hence, α does not depend on the action space size, and for these cases the quality of our prediction will be the same as before. Therefore, for all points in the feature space, the probability of making a correct prediction is either the same or worse when $\varrho < \infty$ than when $\varrho \rightarrow \infty$.

However, that does not complete the proof yet, because a voting event ξ may map to a different point χ when the action space changes. For instance, the number of agents that agree over a suboptimal action may overpass the number of agents that agree on the optimal action as the action spaces changes from $\varrho \rightarrow \infty$ to $\varrho < \infty$. Therefore, we need to show that our prediction will be strictly better when $\varrho \rightarrow \infty$ irrespective of such mapping.

Hence, let us now study the voting events. As the number of actions decrease, a certain voting event ξ when $\varrho \rightarrow \infty$, will map to a voting event ξ' when $\varrho < \infty$ (where ξ may or may not be equal to ξ'). Let χ and χ' be the corresponding points in the feature

space for ξ and ξ' . Also, let \mathcal{H} and \mathcal{H}' be the respective subset of agreeing agents. Let's consider now the four possible cases:

(i) $|\mathcal{H}| = |\mathcal{H}'| = 1$. For such events, the performance of the predictor will remain the same, that is, for both cases we will make a correct prediction with probability $\max(\alpha, 1 - \alpha)$. Note that this case will not happen for all events, as $p_{i,0} \not\rightarrow 0$ when $\varrho \rightarrow \infty$, hence there will be at least one event where $|\mathcal{H}| \geq 2$.

(ii) $|\mathcal{H}| \geq 2, |\mathcal{H}'| \geq 2$. For such events the performance of the predictor will be higher when $\varrho \rightarrow \infty$, as we can make a correct prediction for a point χ with probability 1, while for a point χ' with probability $\max(\beta, 1 - \beta) < 1$.

(iii) $|\mathcal{H}| \geq 2, |\mathcal{H}'| = 1$. This case will not happen under the *ST* agent model. If there was a certain subset \mathbf{H} of agreeing agents when $\varrho \rightarrow \infty$, when we decrease the number of actions the new subset of agreeing agents \mathbf{H}' will either have the same size or will be larger. This follows from the fact that we may have a larger subset agreeing over some suboptimal action when the action space decreases, but the original subset that voted for the optimal action will not change.

(iv) $|\mathcal{H}| = 1, |\mathcal{H}'| \geq 2$. We know that in this case ξ' is an event where the team fails (otherwise the same subset would also have agreed when $\varrho \rightarrow \infty$). Hence, for $1 - \alpha > \alpha$ (weak assumption (i)), we make a correct prediction for such case when $\varrho \rightarrow \infty$. When $\varrho < \infty$, we make a correct prediction if $1 - \beta > \beta$. β , however, depends on the voting profile of the event ξ' . By weak assumption (ii), there will be at least one event where the team is more likely to be correct than wrong (that is, $\beta > 1 - \beta$). Hence, there will be at least one event where our predictor changes from making a correct prediction (when $\varrho \rightarrow \infty$) to making an incorrect prediction (when $\varrho < \infty$).

Hence, for all voting events, the probability of making a correct prediction will either be the same or worse when $\varrho < \infty$ than when $\varrho \rightarrow \infty$, and there will be at least one voting event where it will be worse, completing the proof. Hence, $\varrho \rightarrow \infty$ is strictly the best case for our prediction. As we assume that all world states are independent, if $\varrho \rightarrow \infty$ is the best case for a single world state, it will also be the best case for a set of world states. ■

8.4 Results

8.4.1 Computer Go

I first test my prediction method in the Computer Go domain. I use four different Go software: Fuego 1.1 [Enzenberger et al., 2010], GnuGo 3.8 [FSF, 2009], Pachi 9.01 [Baudiš and Gailly, 2011], MoGo 4 [Gelly et al., 2006], and two (weaker) variants of Fuego (Fuego Δ and Fuego Θ), in a total of six different, publicly available, agents. Fuego is the strongest

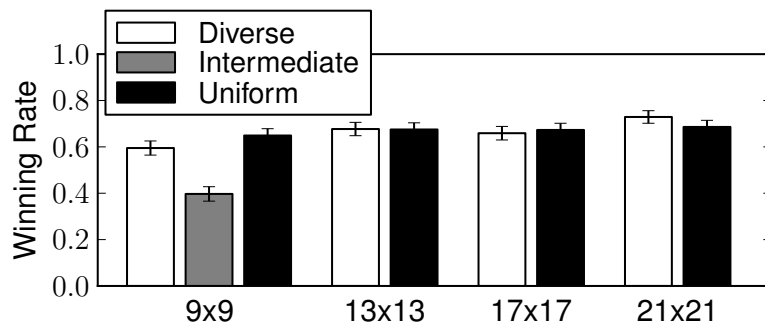


Figure 8.3: Winning rates of the three teams, under four different board sizes.

agent among all of them (Chapter 3). The description of Fuego Δ and Fuego Θ is available in Chapter 3.

I study three different teams: *Diverse*, composed of one copy of each agent¹; *Uniform*, composed of six copies of the original Fuego (initialized with different random seeds, as in Soejima et al. [2010]); *Intermediate*, composed of six random parametrized versions of Fuego (from Chapter 6). In all teams, the agents vote together, playing as white, in a series of Go games against the original Fuego playing as black. I study four different board sizes for *diverse* and *uniform*: 9x9, 13x13, 17x17 and 21x21. For *intermediate*, I study only 9x9, since the random parametrizations of Fuego do not work on larger boards. In Go a player is allowed to place a move at any empty intersection of the board, so the largest number of possible actions at each board size is, respectively: 81, 169, 289, 441.

In order to evaluate my predictions, I use a dataset of 1000 games for each team and board size combination (in a total of 9000 games, all played from the beginning). For all results, I used repeated random sub-sampling validation. I randomly assign 20% of the games for the testing set (and the rest for the training set), keeping approximately the same ratio as the original distribution. The whole process is repeated 100 times. Hence, in all graphs I show the average results, and the error bars show the 99% confidence interval ($p = 0.01$), according to a *t-test*. If the error bars cannot be seen in a certain point in a graph, it is because they are smaller than the symbol used to mark that point in the graph. Moreover, when I say that a certain result is significantly better than another, I mean statistically significantly better, according to a *t-test* where $p < 0.01$, unless I explicitly give a p value.

First, I show the winning rates of the teams in Figure 8.3. This result is not yet evaluating the quality of my prediction, it is merely background information that I will

¹Except for GnuGo, all other agents use Monte Carlo Tree Search. However, each software was developed by different groups, and use different heuristics and implementation strategies. In comparison with the other teams studied, this is the team with the greater diversity.

use when analyzing my prediction results later. On 9x9 Go, *uniform* is better than *diverse* with statistical significance ($p = 0.014$), and both teams are clearly significantly better than *intermediate* ($p < 2.2 \times 10^{-16}$). On 13x13 and 17x17 Go, the difference between *diverse* and *uniform* is not statistically significant ($p = 0.9619$ and 0.5377 , respectively). On 21x21 Go, the *diverse* team is significantly better than *uniform* ($p = 0.03897$).

In order to verify my online predictions, I used the evaluation of the original Fuego, but I give it a time limit $50\times$ longer. I will refer to this version as “Baseline”. I, then, use the Baseline’s evaluation of a given board state to estimate its probability of victory, allowing a comparison with my approach. Considering that an evaluation above 0.5 is “success” and below is “failure”, I compare my predictions with the ones given by the Baseline’s evaluation, at each turn of the games.

I use this method because the likelihood of victory changes dynamically during a game. That is, a team could be in a winning position at a certain stage, after making several good moves, but suddenly change to a losing position after committing a mistake. Similarly, a team could be in a losing position after several good moves from the opponent, but suddenly change to a winning position after the opponent makes a mistake. Therefore, simply comparing with the final outcome of the game would not be a good evaluation. However, for the interested reader, I show in Appendix B how the evaluation would be comparing with the final outcome of the game — and I note here that our prediction quality is still high in such alternative.

Since the games have different lengths, I divide all games in 20 stages, and show the average evaluation of each stage, in order to be able to compare the evaluation across all games uniformly. Therefore, a stage is defined as a small set of turns (on average, 1.35 ± 0.32 turns in 9×9 ; 2.76 ± 0.53 in 13×13 ; 4.70 ± 0.79 in 17×17 ; 7.85 ± 0.87 in 21×21). For all games, I also skip the first 4 moves, since my baseline returns corrupted information in the beginning of the games.

I will present my results in four different sections. First, I will show the analysis for a fixed threshold (that is, the value ϑ above which the output of my prediction function \hat{f} will be considered “success”). Secondly, I am going to analyze across different thresholds using receiver operating characteristic (ROC) curves, and the area under such curves (AUC). Thirdly, I will present the results for different board sizes. Then, finally, I will study the performance of the reduced feature vector.

8.4.1.1 Single Threshold Analysis

I start by showing my results for a fixed threshold, since it gives a more intuitive understanding of the quality of my prediction technique. Hence, in these results I will consider that when my prediction function \hat{f} returns a value above 0.5 for a certain game, it will

be classified as “success”, and when the value is below 0.5, it will be classified as “failure”. In this section I also restrict myself to the 9x9 Go case.

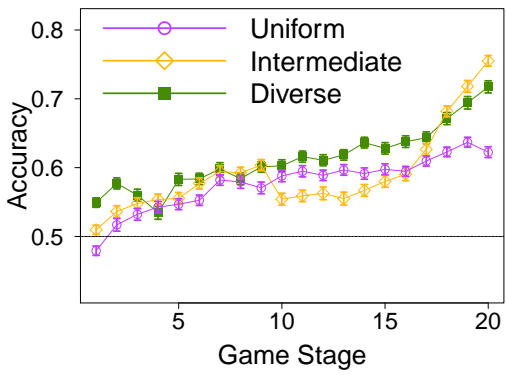
I will evaluate my prediction results according to five different metrics: Accuracy, Failure Precision, Success Precision, Failure Recall, Success Recall. Accuracy is defined as the sum of the true positives and true negatives, divided by the total number of tests (i.e., true positives, true negatives, false positives, false negatives). Hence, it gives an overall view of the quality of the classification. Precision gives the percentage of data points classified with a certain label (“success” or “failure”) that are correctly labeled. Recall denotes the percentage of data points that truly pertain to a certain label, that are correctly classified.

I show the results in Figure 8.4. As we can see, we were able to obtain a high-accuracy very quickly, already crossing the 0.5 line in the 2nd stage for all teams. In fact, the accuracy is significantly higher than the 0.5 mark for all teams after the 2nd stage (and for *diverse* and *intermediate* since the 1st stage).

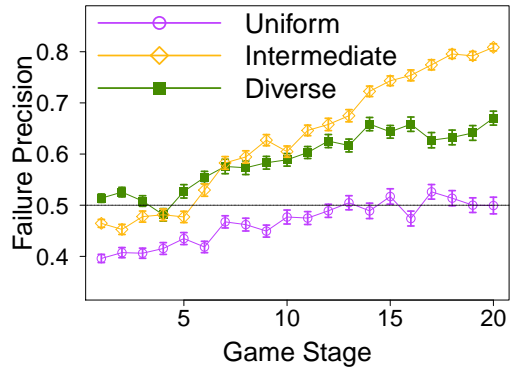
From around the middle of the games (stage 10), the accuracy for *diverse* and *uniform* already gets close to 60% (with *intermediate* only close behind). Although we can see some small drops – that could be explained by the sudden changes in the game –, overall the accuracy increases with the game stage number, as expected. Moreover, for most of the stages, the accuracy is higher for *diverse* than for *uniform*. The prediction for *diverse* is significantly better than for *uniform* in 90% of the stages. It is also interesting to note that the prediction for *intermediate* is significantly better than for *uniform* in 60% of the stages, even though *intermediate* is a significantly weaker team. In fact, we can see that in the last stage, the accuracy, the failure precision and the failure recall is significantly better for *intermediate* than for the other teams.

8.4.1.2 Multiple Threshold Analysis

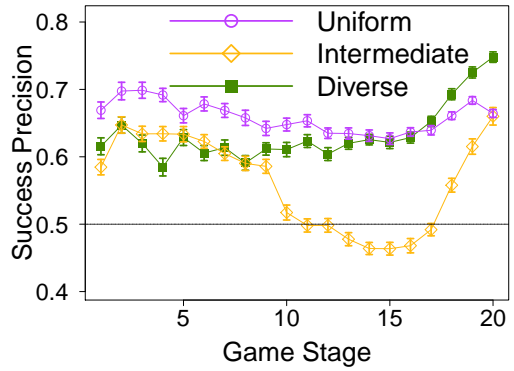
I now measure my results under a variety of thresholds (that is, the value ϑ above which the output of our prediction function \hat{f} will be considered “success”). In order to perform this study, I use receiver operating characteristic (ROC) curves. An ROC curve shows the true positive and the false positive rates of a binary classifier at different thresholds. The true positive rate is the number of true positives divided by the sum of true positives and false negatives (i.e., the total number of items of a given label). That is, the true positive rate shows the percentage of “success” cases that are correctly labeled as such. The false positive rate, on the other hand, is the number of false positives divided by the sum of false positives and true negatives (i.e., the total number of items that *are not* of a given label). That is, the false positive rate shows the percentage of “failure” cases that are wrongly classified as “success”.



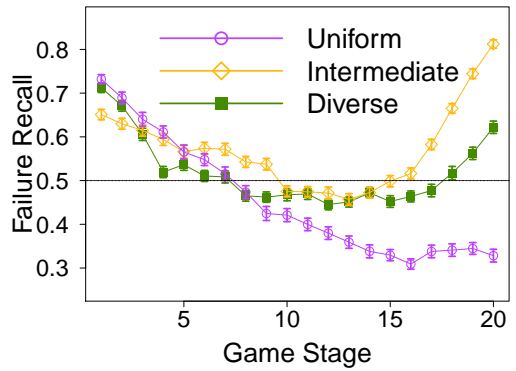
(a) Accuracy



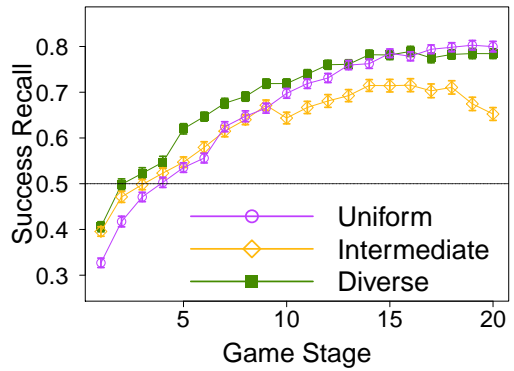
(b) Failure Precision



(c) Success Precision



(d) Failure Recall



(e) Success Recall

Figure 8.4: Performance metrics over all turns of 9x9 Go games.

Hence, ROC curves allow us to better understand the trade-off between the true positives and the false positives of a given classifier. By varying the threshold, we can increase the number of items that receive a given label, increasing henceforth the number of items of such label that are correctly classified, but at the cost of also increasing the number of items that incorrectly receive the same label.

The ideal classifier is the one on the upper left corner of the ROC graph. Such point (true positive rate = 1, false positive rate = 0) indicates that every item classified with a certain label truly pertains to such label, and every item that truly pertains to such label receives the correct classification.

In this chapter, I do not aim only on studying the performance of a single classifier, but rather on comparing the performance of our prediction technique on a variety of situations, changing the team and the action space size (i.e., the board size). Hence, I also study the area under the ROC curve (AUC), as a way to synthesize the quality information from the curve into a single number. The higher the AUC, the better the prediction quality of my technique in a given situation. That is, a completely random classifier would have an AUC of 0.5, and as the ROC curve moves towards the top-left corner of the graph, the AUC gets closer and closer to 1.0.

In fact, the AUC metric has been shown to be equal to the probability of, given two pair of items with different labels (one “success” case and one “failure” case, in our situation), correctly considering the item that was truly a “success” as more likely to be a “success” case than the other item. It has also been shown to be related to other important statistical metrics, such as the *Wilcoxon test of ranks*, the *Mann-Whitney U* and the *Gini coefficient* [Hanley and McNeil, 1982, Mason and Graham, 2002, Hand and Till, 2001].

I start by studying multiple thresholds in a fixed board size (9x9), and in the next section I study the effect of increasing the action space. Hence, in Figure 8.5 I show the ROC curves for all teams in the 9x9 Go games, for 4 different stages of the game. Although in the beginning of the game (stage 5) it is hard to distinguish the results for the different teams, we can note that from the middle of the games to the end (stage 15 and 20), there is a clear distinction between the prediction quality for *diverse* and *intermediate*, when compared with the one for *uniform*, across many different thresholds.

As mentioned, to formally compare these results across all stages I use the AUC metric. Hence, in Figure 8.6, I show the AUC for the three different teams in 9x9 Go. As we can see, the three teams have similar AUCs up to stage 10, but from that stage on we can get better AUCs for both *diverse* and *intermediate*, significantly outperforming the AUC for *uniform* in all stages. We also find that, considering all stages, we have a significantly better AUC for *diverse* (than for *uniform*) in 85% of the cases, and for *intermediate*

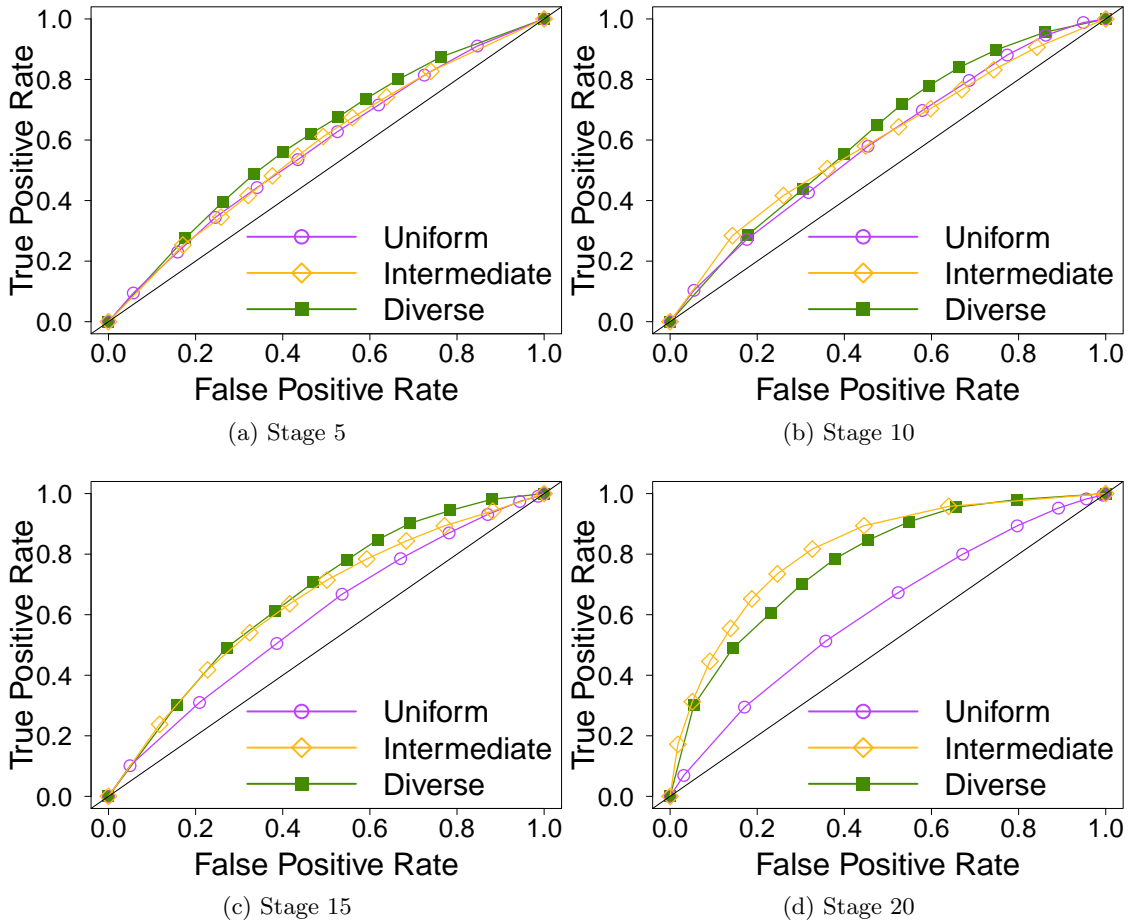


Figure 8.5: ROC curves, analyzing different thresholds in 9x9 Go.

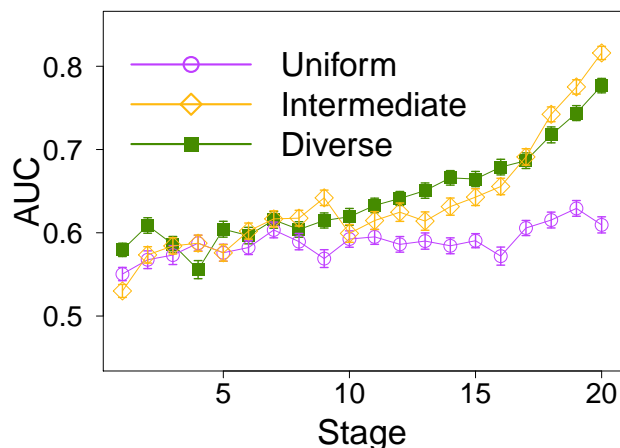


Figure 8.6: AUC for *diverse*, *uniform* and *intermediate* teams, in 9x9 Go.

in 55% of the cases. Curiously, we can also note that even though *intermediate* is the weakest team, we can obtain for it the (significantly) best AUC in the last stage of the games, surpassing the AUC found for the other teams. Overall, these results show that my hypothesis that we can make better predictions for teams that have higher diversity holds irrespective of the threshold used for classification. On the next section, I study the effect of increasing the action space size.

8.4.1.3 Action Space Size

Let us start by showing, in Figure 8.7, ROC curves for *diverse* and *uniform* under different board sizes. It is harder to distinguish the curves on stages 5 and 10, but we can notice that the curve for 21x21 tends to dominate the others on stages 15 and 20. Again, to better study these results, we look at how the AUC changes for different teams and board sizes.

In Figure 8.8 we can see the AUC results. For the *diverse* team (Figure 8.8 (a)), we start observing the effect of increasing the action space after stage 5, when the curves for 17×17 and 21×21 tends to dominate the other curves. In fact, the AUC for 17×17 is significantly better than smaller boards in 60% of the stages, and in 80% of the stages after stage 5. Moreover, after stage 5 no smaller board is significantly better than 17×17 . Concerning 21×21 , we can see that from stage 14, its curve completely dominates all the other curves. In all stages from 14 to 20 the result for 21×21 is significantly better than for all other smaller boards. Hence, we can note that the effect of increasing the action space seems to depend on the stage of the game, and it gets more evident as the stage number increases.

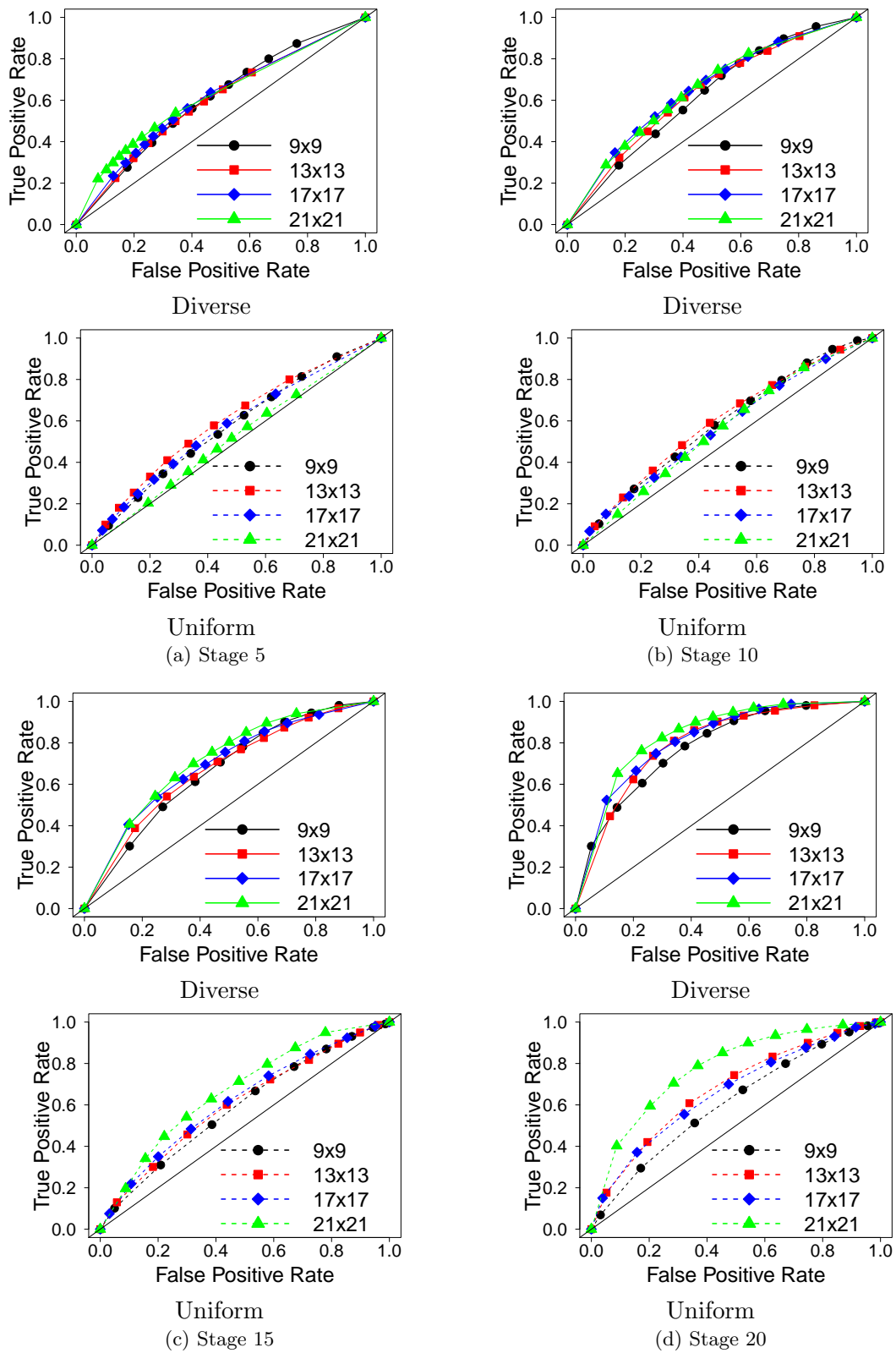


Figure 8.7: ROC curves for *diverse* and *uniform*, for different board sizes.

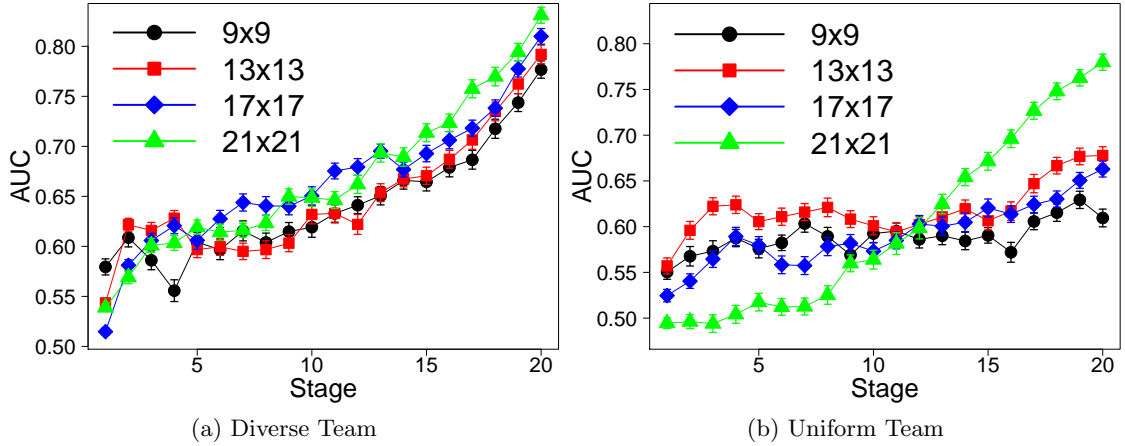


Figure 8.8: AUC for different teams and board sizes, organized by teams.

Concerning the *uniform* team (Figure 8.8 (b)), up to 17×17 we cannot observe a positive impact of the action space size on the prediction quality; but for 21×21 there is clearly an improvement from the middle game when compared with smaller boards. On all 8 stages from stage 13 to stage 20, the result for 21×21 is significantly better than for other board sizes.

In terms of percentage of stages where the result for 21×21 is significantly better than for 9×9 , we find that it is 40% for the *uniform* team, while it is 85% for the *diverse* team. Hence, the impact of increasing the action space occurs for *diverse* earlier in the game, and over a larger number of stages.

Now, in order to compare the performance for *diverse* and *uniform* under different board sizes, I show the AUCs in Figure 8.9 organized by the size of the board (Figure 8.6 is repeated here in Figure 8.9 (a) to make it easier to observe the difference between board sizes). It is interesting to observe that the quality of the predictions for *diverse* is better than for *uniform*, irrespective of the size of the action space. Moreover, while for 9×9 and 13×13 the prediction for *diverse* is only always significantly better than for *uniform* after around stage 10, we can notice that for 17×17 and 21×21 , the prediction for *diverse* is always significantly better than for *uniform*, irrespective of the stage (except for stage 1 in 17×17). In fact, I can also show that the difference between the teams is greater on larger boards. In Figure 8.10 we can see the difference between *diverse* and *uniform*, in terms of area under the AUC graph, and also in terms of percentage of stages where *diverse* is significantly better than *uniform*, for 9×9 and 21×21 . The difference between the areas in 9×9 and 21×21 is statistically significant, with $p = 0.0003337$.

I study again the accuracy, precision and recall in 21×21 , as the results in these metrics may be more intuitive to understand for most readers (although limited to a fixed

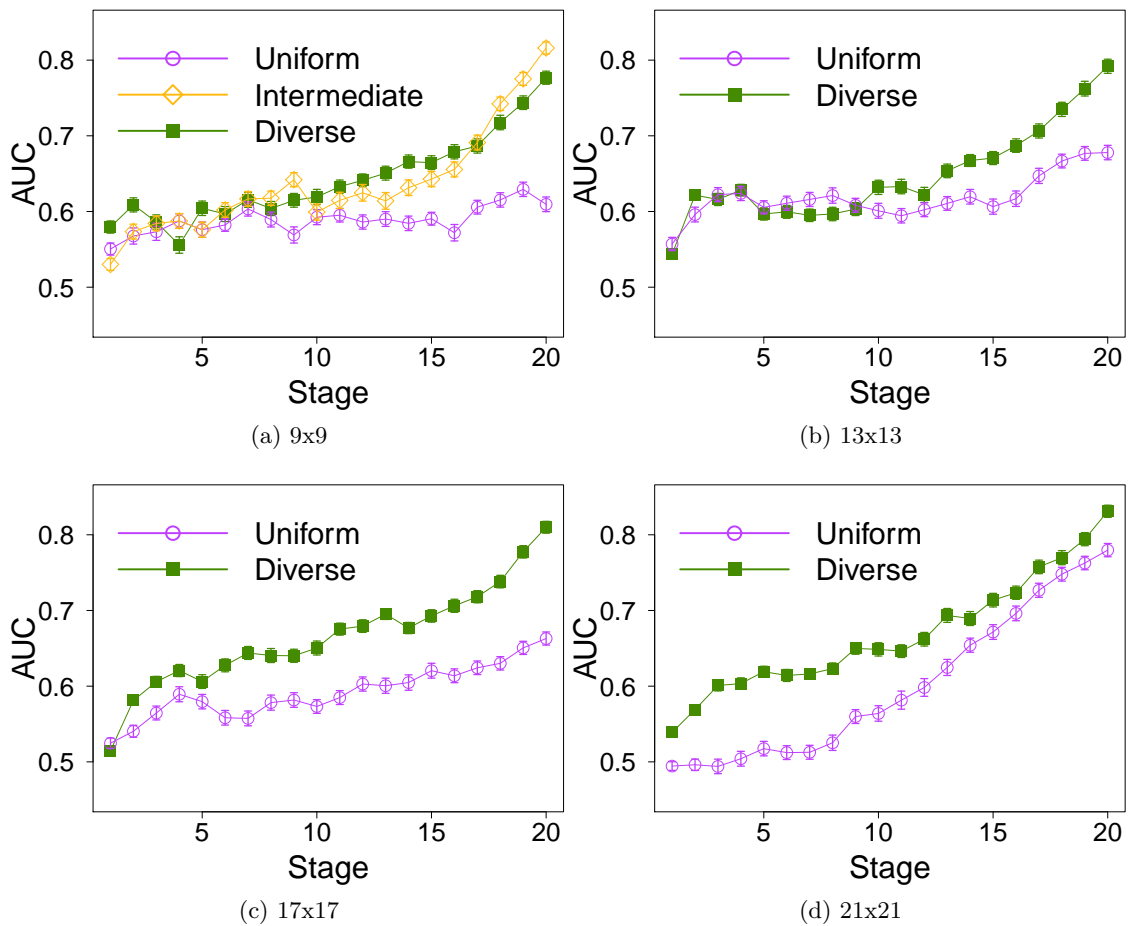


Figure 8.9: AUC for different teams and board sizes, organized by board sizes.

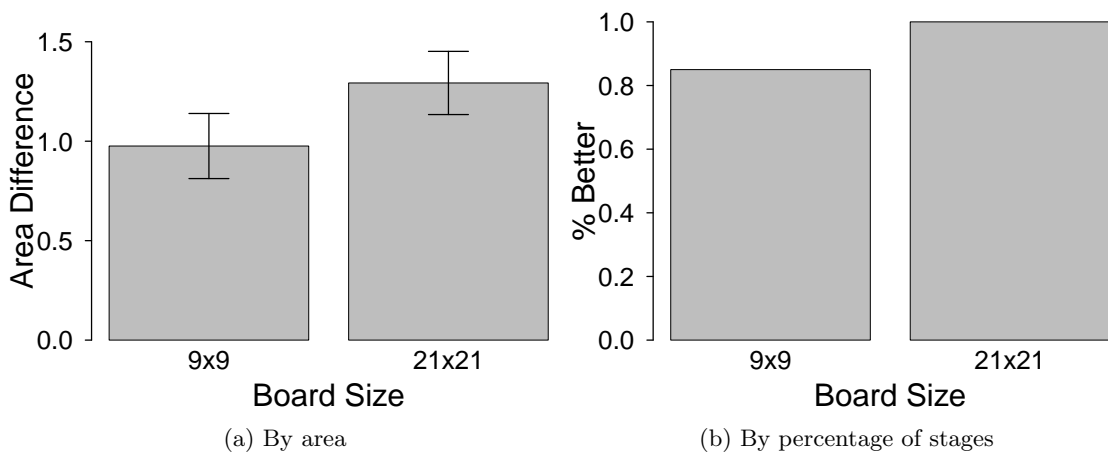


Figure 8.10: Differences in prediction quality for the *diverse* and *uniform* teams.

threshold, in this case 0.5). I show these results in Figure 8.11, where for comparison I also plot the results for 9×9 . For the *diverse* team, the accuracy in 21×21 is consistently significantly better than in 9×9 , in all stages from stage 12. We also have a significantly better accuracy in the first 5 stages. Concerning *uniform*, the accuracy for 21×21 is only consistently better from stage 15 (and also in stage 1). Hence, again, we can notice a better prediction quality on larger boards, and also a higher impact of increasing the action space size on *diverse* than on *uniform* (in terms of number of stages, and also how early the performance improves).

8.4.1.4 Reduced Feature Vector

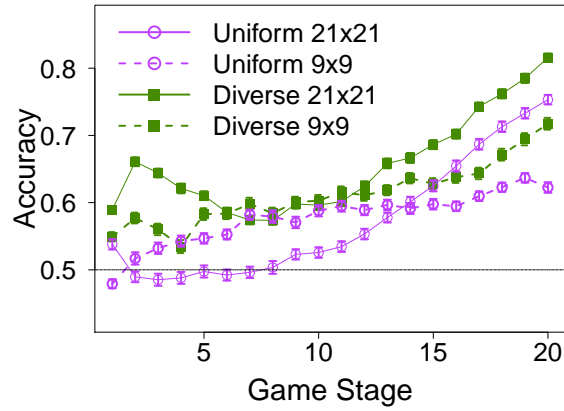
Finally, I study the reduced feature vector. The results are very similar to the ones using the full feature vector, so I do not repeat them here. I note, however, that we can obtain similar results with a much more scalable representation². In fact, in Figure 8.12 I study the area under the AUC graphs of the full and reduced representations, for all combinations of teams and board sizes. Surprisingly, the reduced representation turns out to be significantly better for all teams on the 9×9 and 13×13 boards. This may happen since it is easier to learn a model with less features. For the *diverse* team, however, the importance of the full representation increases as the action space grows. On 17×17 , the difference between the representations is not statistically significant ($p = 0.9156$), while on 21×21 the full representation is actually significantly better than the reduced one ($p = 0.001794$).

8.4.2 Ensemble Learning

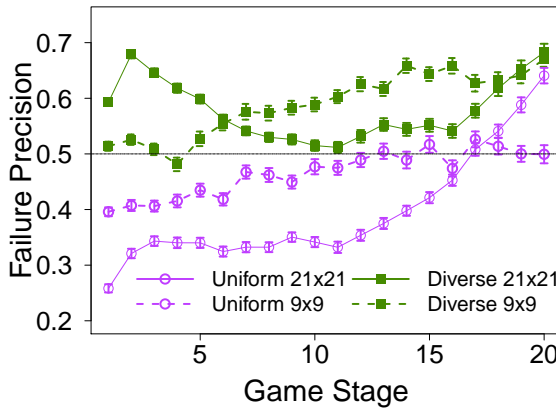
In this section, I demonstrate that my approach also applies to other domains, by using my technique in order to predict the performance of an ensemble system. Note that I am not using here an ensemble system to predict the performance of a team. I am still using a single predictor function, but now the team of agents whose final performance we want to predict is a set of classifiers voting in order to assign labels to sets of items. Hence, an agent here corresponds to one classifier, an action corresponds to a label assignment, and a world state corresponds to an item.

I use the scikit-learn’s digits dataset [Pedregosa et al., 2011], which is composed of 1797 8×8 images. Each image is a hand-written digit, and the objective of the ensemble is to correctly identify the digit. Hence, for each item (i.e., image), there are 10 possible labels, corresponding to each possible digit, and only one of these labels will be a correct

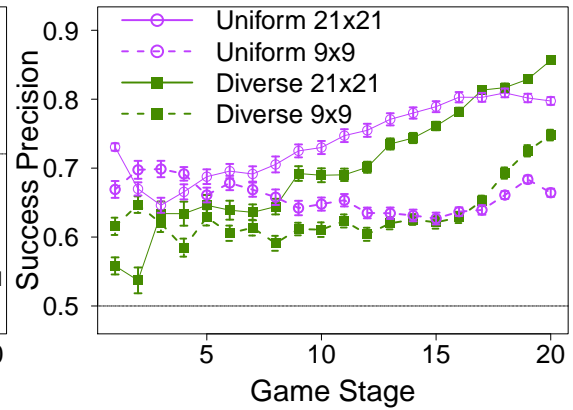
²All my main conclusions still hold, except that this time the AUC for 21×21 is significantly better than for 9×9 around stage 15 for both teams.



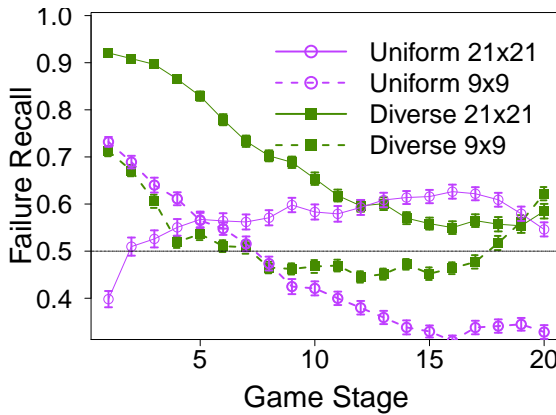
(a) Accuracy



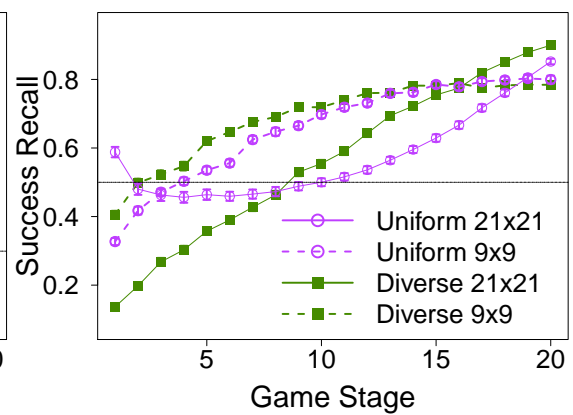
(b) Failure Precision



(c) Success Precision



(d) Failure Recall



(e) Success Recall

Figure 8.11: Performance metrics over all turns of 9x9 and 21x21 Go games.

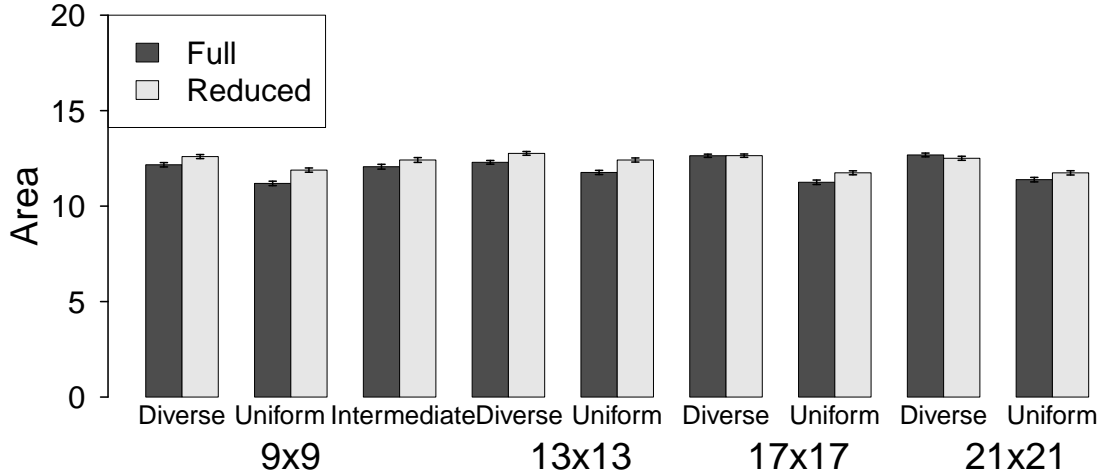


Figure 8.12: Comparison of prediction quality with the full and reduced representation.

classification. I randomly select 1000 items for training the agents. Among the chosen 1000 samples, 400 randomly selected items are used in training each individual agent. I use the remaining 797 items for the testing phase, where the agents will vote to assign labels to sets of 50 items randomly sampled from the remaining 797. To avoid confusion, note that I am referring here to the training process of the individual agents, not of my team prediction methodology (which I will discuss later in this section).

I use 6 classification algorithms: SVM, Logistic Regression, Decision Tree, Nearest Neighbour, Bernoulli Naive Bayes, Gaussian Naive Bayes. Each classification algorithm corresponds to one agent. I study two different teams. *Diverse 1*, composed by copies of the SVM agent with different parametrizations; and *Diverse 2*, composed by one copy of each agent. Unlike in the Go domain, where multiple copies of the same agent (like Fuego), would still vote for different actions due to the randomness involved in the rollouts of the UCT Monte Carlo algorithm, a classifier is deterministic given its algorithm and parametrization. Therefore, I do not present results for *Uniform* in this domain, and instead I present two possible variations for a diverse team.

I define one instance of the problem solving process as a set of 50 items to be classified. Similar to the previous domain, where we have a large set of games (each with a set of turns), we will have here multiple sets of 50 items to be classified (each corresponding to one world state). Each set of items is constructed from random sampling 50 items from our testing set (i.e., the 797 items that *were not* used to train the agents).

The agents will vote to assign labels for each item, and the problem solving process will be completed when the team assigns a label to each one of the 50 items. If the team classifies correctly more than a threshold percentage of the items, I consider it a

“Success”. Otherwise, the performance of the team is classified as “Failure”. I explain below how the threshold is calculated.

I randomly generate 2500 sets in this fashion. I will now discuss the training of my team prediction methodology. I randomly select 2000 sets for training, where the agents voted to classify each item, as discussed (note that the training set previously discussed in this section was used to train the individual agents). I use the mean performance over these 2000 sets as our threshold. Hence, if the team is able to classify a higher number of items than its average performance, I consider it a “success”. Otherwise, the problem solving process will be classified as “failure”.

The remaining sets are used to test my team prediction methodology. I again evaluate accuracy, precision and recall. One run consists of the ensemble evaluating one set of items, and the performance metric at each stage will be calculated as the average over 500 runs (similarly as in the Computer Go case, where I evaluated across multiple games).

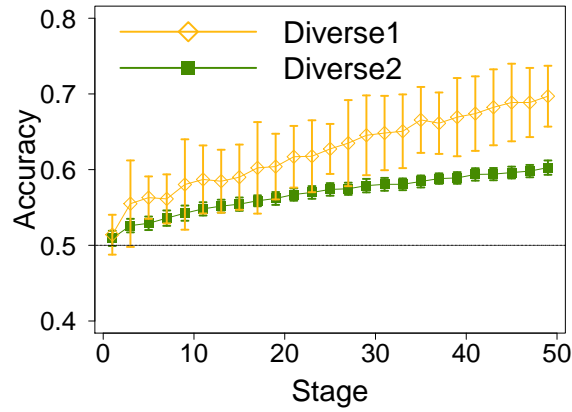
As before, I use random sub-sampling evaluation. Hence, the whole training and testing procedure is repeated 50 times; and at each time the training and the testing set (of my team prediction methodology) are randomly sampled from the full database of runs. Again, in all graphs I show the average results, and the error bars show the 99% confidence interval ($p = 0.01$), according to a *t-test*. When I say that a certain result is significantly better than another, I mean statistically significantly better, according to a *t-test* where $p < 0.01$, unless I explicitly give a p value.

I show the results in Figure 8.13, where each stage corresponds to an item to be classified. As we can see, we are able to obtain high quality predictions for both teams quickly. The accuracy for both teams is always significantly higher than 50% after Stage 3. Around the middle of the runs (i.e., Stage 25), the accuracy of *Diverse 1* is already around 60%, with *Diverse 2* only close behind. Towards the end of the runs we obtain around 70% accuracy for *Diverse 1* and 60% accuracy for *Diverse 2*. Note that we also obtain results significantly better than 50% for most of the stages in the other performance metrics.

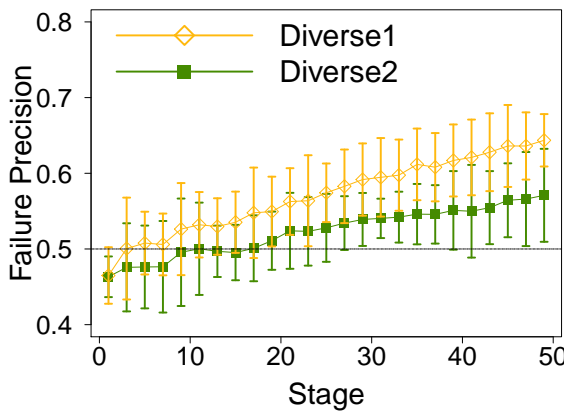
In Figure 8.14, we can see the results using the reduced feature vector. As we can see, we can still obtain high-quality predictions in a more compact representation. This holds especially for *Diverse 1*, as we obtain an accuracy higher than 70% towards the end of the runs.

8.5 Analysis of Coefficients

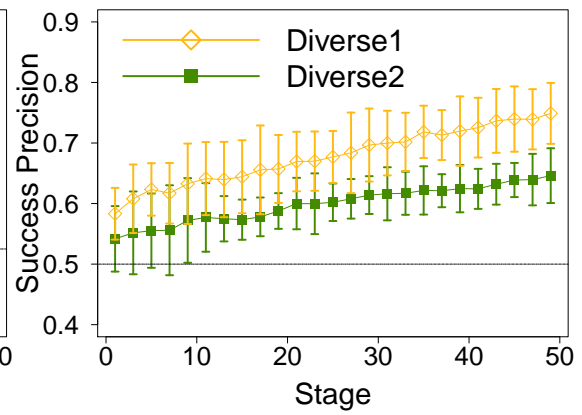
I analyze the coefficients that are learned in my experiments, in order to better understand the prediction functions. I focus here in analyzing the reduced feature vector in



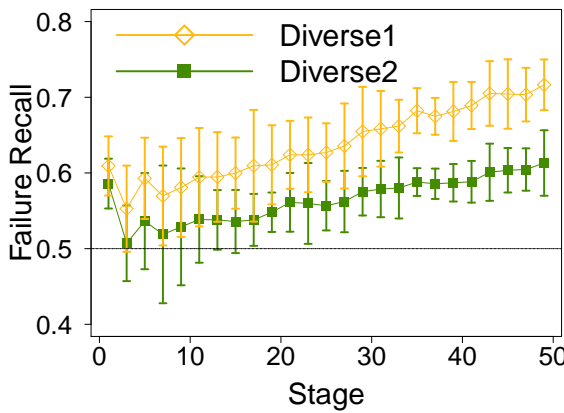
(a) Accuracy



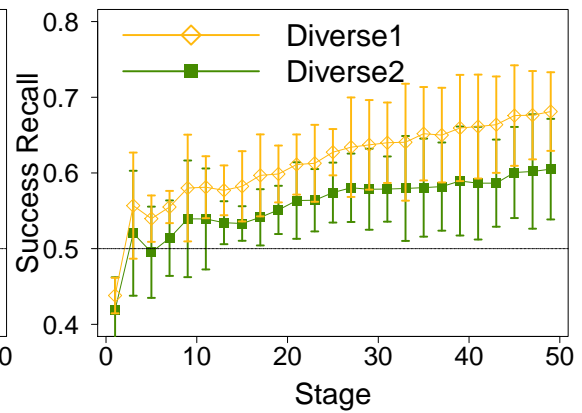
(b) Failure Precision



(c) Success Precision

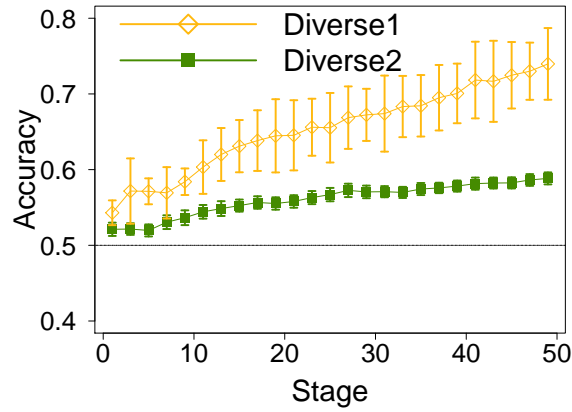


(d) Failure Recall

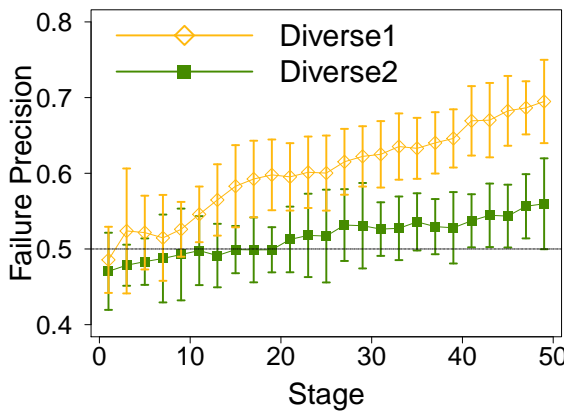


(e) Success Recall

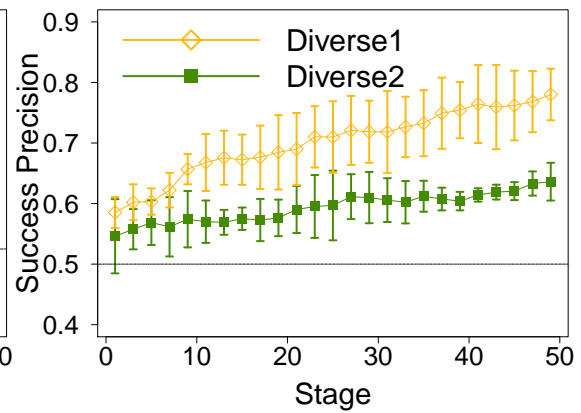
Figure 8.13: Performance metrics over a set of items in ensemble system classification (full feature vector).



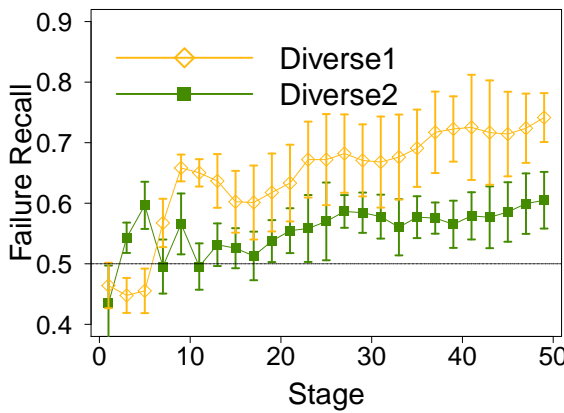
(a) Accuracy



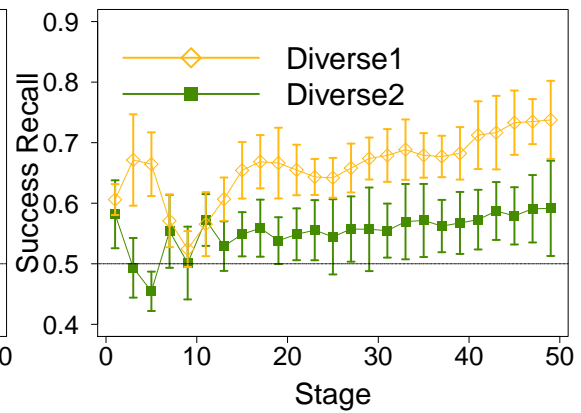
(b) Failure Precision



(c) Success Precision



(d) Failure Recall



(e) Success Recall

Figure 8.14: Performance metrics over a set of items in ensemble system classification (reduced feature vector).

the Computer Go domain, as its lower number of coefficients makes the analysis more practicable (and its results are similar to the ones with the full feature vector). Hence, I study the different coefficients learned for each subset size (i.e., the size of the subset of agents that agrees on the chosen action), under different teams and board sizes. As a way to make a comparison among the coefficients under different situations, I plot the normalized coefficients, by dividing each by their overall sum. This allows us to study how the relative importance of each coefficient in determining whether a team will be successful changes under different situations. Similarly to the previous section, I also plot error bars showing the 99% confidence interval (i.e., $p = 0.01$) by performing a t -test over the results of my random subsampling validation.

I start by plotting, in Figure 8.15, the coefficients organized according to the board sizes, to better compare how they change for each team. Note that the closer to 0 a coefficient is (i.e., the lower the bar size in the graphs), the higher the importance of the corresponding subset size in determining that a game will be a victory for the team. Also, the numbers on the x axis indicate the respective subset size.

We can make several observations from these graphs. First, note that in general the learned coefficients are quite stable, as the low error bars indicate a low variance across different samples. Second, for almost all the coefficients we have a statistically significant difference across the different teams. This indicates that the learned functions are not trivial, as the appropriate values for each coefficient strongly depends on the team members.

Furthermore, we normally would expect the coefficients to increase (i.e., get closer to 0) as the respective subset size increases, capturing the notion that the higher the agreement, the higher the likelihood of success. However, that is not always the case, and we can, in fact, observe many surprising coefficients, especially in larger board sizes. On 21×21 Go, the “strict increase” hypothesis does not happen for both the *diverse* and the *uniform* teams. The coefficient for subsets of size 1 for the *uniform* team is the most surprising result, as it seems to indicate some tendency of winning the games when the agents fully disagree. For *intermediate*, we can notice such non-strict increase even on 9×9 Go (as the coefficient for subsets of size 4 is significantly higher than the one for subsets of size 5), which may be caused by this being the weakest team (and, hence, more agents agreeing on the same action may not really indicate a higher likelihood of the action being optimal).

In order to better analyze how the coefficients change as the board size increases, I plot in Figure 8.16 the coefficients organized by teams. It is interesting to note that the coefficients evolve in different ways for the different teams. For *diverse*, we notice an increase in the importance of subsets of size 3 in 21×21 Go, curiously significantly

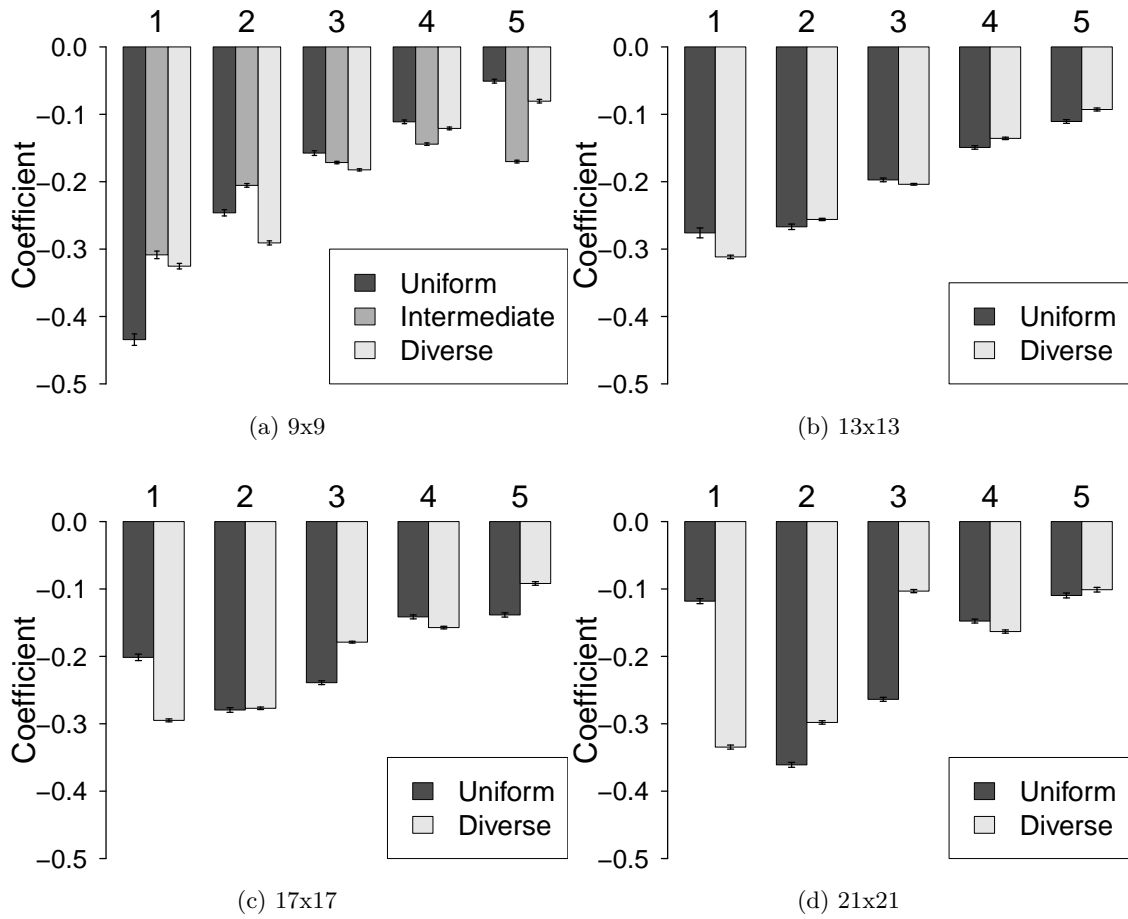


Figure 8.15: Normalized coefficients of all teams and board sizes, organized by board sizes.

surpassing subsets of size 4. One possible explanation could be that in such board size the team is more likely correct when the three strongest agents are in (exclusive) agreement.

For the *uniform* team the results are even more surprising. We can notice a consistent increase in the importance of subsets of size 1. This seems to indicate that the team is more likely to be winning when the team is in full disagreement as the board size grows, which does not correspond well with the *ST* agent model. The *ST* agent model, however, was originally created to model diverse teams (Chapter 4), hence a deviation on uniform teams could be expected. Perhaps this result may happen because the games where the agents fully disagree tend to be more complex, and the *uniform* team may be in a higher advantage on dealing with such complex positions than its opponent.

It is worthy of notice that even though the prediction functions for these teams are so different, and evolve in such different ways as the number of actions increases, we could still learn them adequately for all these different situations, always obtaining high-quality prediction results. This shows that my learning methodology is robust, as it is able to learn good prediction functions for a variety of situations.

8.6 Discussion

I show, both theoretically and experimentally, that we can make high-quality predictions about the performance of a team of voting agents, using only information about the frequency of agreements among them. I present two kinds of feature vectors, one that includes information about which specific agents were involved in an agreement and one that only uses information about how many agents agreed. Although the number of features in the former increases exponentially with the number of agents, causing scalability concerns, the latter representation scales much better, as it increases linearly. Theoretically, the full feature vector should have better results in general, but as I discuss in Corollary 8.3.5, the reduced representation approximates well the full one under some conditions. In fact, however, in my experiments we find that, although the results are similar, the reduced representation is statistically significantly better in most cases. This may happen since it is easier to learn a model with less features. Hence, for large teams we can safely use the reduced feature vector, avoiding scalability problems.

Moreover, in real applications we usually do not have extremely large teams of voting agents. Unless we have an “idealized” diverse team, the performance is expected to converge after a certain number of agents (Chapter 6). In Chapters 3 and 4, significant improvements are already obtained with only 6 agents, while Chapter 6 shows little improvement as teams grow larger than 15 agents. Therefore, even in cases where Corollary

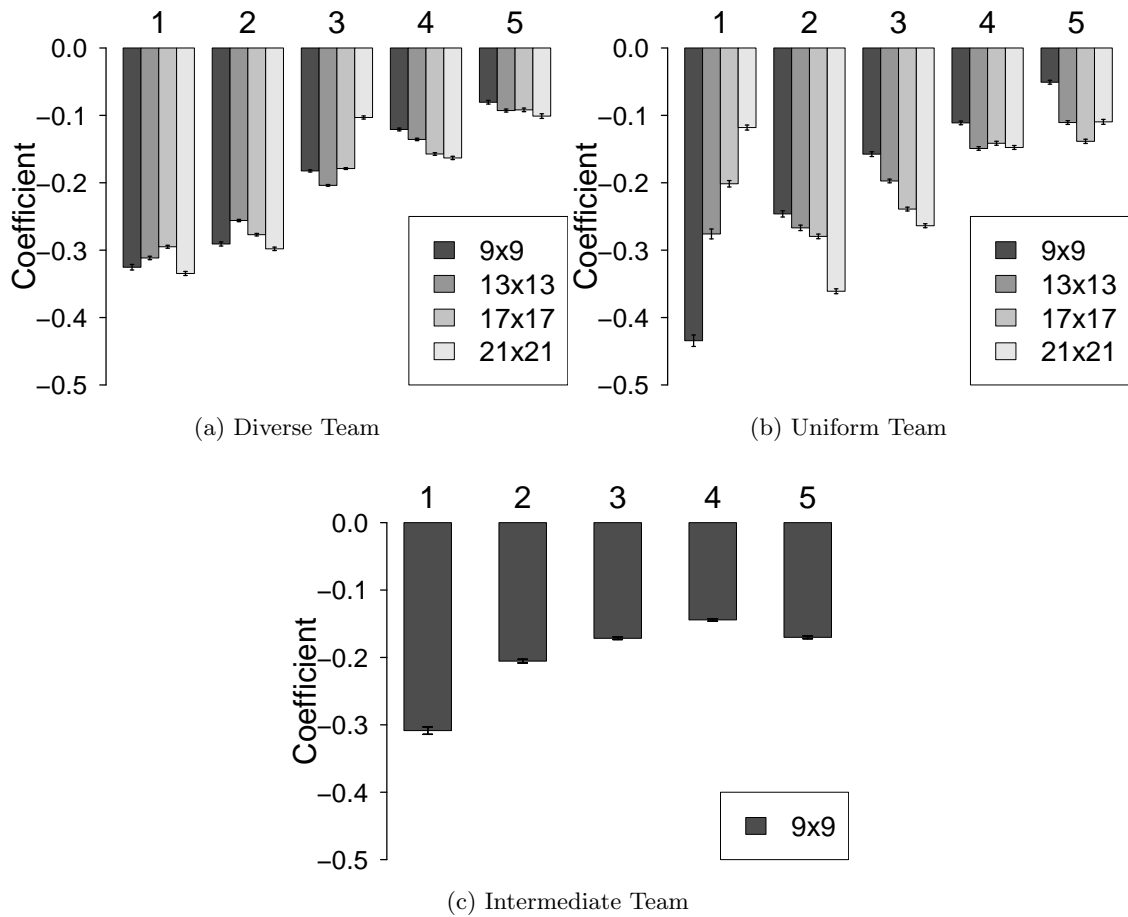


Figure 8.16: Normalized coefficients of all teams and board sizes, organized by teams.

8.3.5 does not apply, and the reduced vector is not a good approximation, the scalability of the full feature vector might not be a real concern.

Based on classical voting theory, and in my Proposition 8.3.1, Observation 8.3.2 and Proposition 8.3.3, we would expect the predictions to work better for the *uniform* team, if it is composed of copies of the best agent. However, I present a more general model in Theorem 8.3.4, that does not really depend on plurality being a maximum likelihood estimator (MLE). In fact, I show in my experiments that the prediction works significantly better for the *diverse* team, and I explain this phenomenon in Corollary 8.3.6. Moreover, the prediction for *intermediate* works as well as for the other teams, and also significantly better than for the other teams towards the last stages of the games, even though it is a significantly weaker team. We would not expect this result, based on classical voting theory and Proposition 8.3.1, Observation 8.3.2 and Proposition 8.3.3. We can, however, expect such result based on my more general model in Theorem 8.3.4, as I show that the prediction does not really depend at all on plurality being a MLE. Hence, it can still work well in cases where the team is weaker, and the MLE assumption does not hold well.

We also observed that the quality of my prediction increases significantly as the board size increases, for both the *diverse* and the *uniform* team, but the impact for *diverse* occurs earlier in the game, and over a larger number of stages. I explain such phenomenon in Theorem 8.3.7. As normally complex problems are characterized by having a large action space, this shows that we can make better predictions for harder problems, when it is actually more useful to be able to make predictions about the team performance.

As I study the *Receiver Operating Characteristic* (ROC) curves, and the respective Areas Under the Curves (AUC), I can also show that my experimental conclusions hold irrespective of how we interpret the output of the prediction function. Hence, we find that across many different thresholds we can still make better predictions for the *diverse* team, and the prediction quality still improves in larger boards. We also notice that the difference on the prediction quality for *diverse* and *uniform* teams increases in larger boards.

I analyze the coefficients of the reduced feature vector, and note that they are highly non-trivial. The way the coefficient for subsets of size 1 evolves for the *uniform* team is the most surprising result, and seems to suggest that the performance improves for such team for a different reason than the one presented in Theorem 8.3.7. As the theorem assumes *ST* agents, however, we can expect it to cover better the situation of the *diverse* team. A better understanding of why the coefficients turned out to be how they are is an interesting direction for future work. Nevertheless, it is a positive result that my learning technique could work well and learn adequate prediction functions across all these different situations.

I also applied my technique to the Ensemble Learning domain, where I was able to predict the final performance of a team of classifiers in a dataset of hand-written digits recognition. This shows that my model is really domain independent, as it worked well in two completely different domains.

However, although I showed a great performance in prediction, I did not present in this chapter what an operator should actually do as the prediction of failure goes high. Possible remedial procedures (and the relative qualities of each one of them) vary according to each domain, but here I discuss some possible situations.

For example, consider a complex problem being solved in a cluster of computers. We do not want to overly allocate resources, as that would be a waste of computational power that could be allocated to other tasks (or a waste of electrical energy, at the very least). However, we could increase the allocation of resources for solving the current problem when it becomes necessary, according to our prediction.

While playing a game, it is well known that the best strategy changes according to each specific opponent we are facing [Ganzfried and Sandholm, 2011, Southey et al., 2005, Lockett et al., 2007, Schadd et al., 2007, Bakkes et al., 2009]. However, it is in general hard to know which player is our current antagonist. Therefore, we could start playing the game with the team that works the best against general opponents, and dynamically change the team as our prediction of failure goes high, trying to adapt to the current situation.

Moreover, it is known that the best voting rule depends on the noise model of the agents [Conitzer and Sandholm, 2005]. However, in general, such model is not known for existing agents (Chapter 3). Therefore, we could start by playing a game with a very general rule, such as plurality voting, and dynamically try different voting rules according to our current prediction. Note that although Proposition 8.3.1, Observation 8.3.2 and Proposition 8.3.3 refer to plurality voting, my general model presented in Theorem 8.3.4 does not really depend on the voting rule.

Finally, in this chapter I always trained and tested my classifier under similar conditions, as in traditional machine learning approaches. Training and testing under different conditions leads to a transfer learning problem, which is a current topic of research in the machine learning literature [Banerjee and Stone, 2007, Konidaris et al., 2012, Taylor and Stone, 2009]. Hence, exploring transfer learning in the context of my technique is an interesting direction for future work.

8.7 Conclusion

Voting is a widely applied domain independent technique, that has been used in a variety of domains, such as: machine learning, crowdsourcing, board games, forecasting systems, etc. In this chapter, I present a novel method to predict the performance of a team of agents that vote together at every step of a complex problem. My method does not use any domain knowledge and is based only on the frequencies of agreement among the agents.

I explain theoretically why my prediction works. First, I present an explanation based on classical voting theories, but such explanation fails to fully explain my experimental results. Hence, I also present a more general model, that is independent of the voting rule, and also independent of any optimality assumptions about the voting rule (i.e., the voting rule does not need to be a maximum likelihood estimator across all world states). Such model allows a deeper understanding of the prediction methodology, and a more complete comprehension of my experimental results.

I perform experiments in the Computer Go domain with three different teams, each having different levels of diversity and strength (i.e., performance), and four different board sizes. I showed that the prediction works online at each step of the problem solving process, and matches often with an in-depth analysis (that takes orders of magnitude longer time). Hence, I could achieve a high prediction quality for all teams, despite their differences. In particular, I show that, contrary to what would be expected based on classical voting models, my prediction is not better for stronger teams, and can actually work equally well (even significantly better in some cases) for a very weak team. Moreover, I showed that I could achieve a higher prediction quality for a diverse team than for a uniform team. Furthermore, I verified experimentally that the prediction quality increase as the action space (board size) grows (and the impact occurs earlier in the game for the diverse team). All that could be expected based on my more general model.

I also study the *Receiver Operating Characteristic* curves of my results, and their respective Areas Under the Curves (AUC). Hence, I verified that my conclusions hold across many different thresholds (i.e., different ways to interpret the output of my prediction function).

Moreover, I tested my approach when predicting the final performance of a team of classifiers, a domain that is very different than Computer Go. I was still able to obtain high-quality predictions, showing that my approach really works across different domains.

Finally, I studied in detail the prediction functions learned. My analysis showed that these functions are highly non-trivial, and vary significantly according to each team and board size. In fact, better understanding how and why these coefficients evolved in the

way they did as the board size grows for different teams is an open direction for further study. Overall, however, my prediction technique is very robust, as it could learn adequate functions across all these situations.

Therefore, a system operator can use my technique to take remedial procedures if the team is not performing well, or incorporated within an automatic procedure to dynamically change the team and/or the voting rule. I discussed in detail how that could be executed in different domains.

Hence, this chapter is a significant step towards not only a deeper understanding of voting systems, but also towards a greater applicability of voting in a variety of domains, by combining the potential of voting in finding correct answers with the ability to access the performance of teams and the predictability of the final outcome.

Part V

Discussions and Conclusions

Chapter 9

CONCLUSIONS

This thesis presented theoretical and experimental results in decision-centered teamwork, a novel paradigm in artificial intelligence. Decision-centered teamwork is the analysis of agent teams that iteratively take joint decisions into solving complex problems. I divide it in three fundamental challenges: Agent Selection, Aggregation of Opinions and Team Assessment.

In Agent Selection, my main focus was on the importance of diversity when forming teams. I presented three models: the first shows that a diverse team can outperform a uniform team, and gives the necessary conditions for that to happen. The second shows that diverse teams get stronger in large action spaces, allowing one to better identify when diverse teams should be used. It introduces the novel model of *spreading tail* agents, which assign a non-zero probability to a larger set of actions as the action space increases. The third model studies agent teams for design problems, where the focus is on providing a large set of optimal actions, for a human to select according to aesthetics or other factors. I show that diverse teams improve as the number of agents grows, while uniform teams actually decrease in performance.

In Aggregation of Opinions, I studied ranking extraction techniques, and showed in the Computer Go domain that the Borda voting rule outperforms plurality when using the *ranking by sampling* technique, which ranks actions according to how frequently they are played when sampling an agent multiple times. However, in the context of influencing social networks, weighted plurality was still the best voting rule.

I also studied the simultaneous influencing and mapping problem, where we must iteratively discover a network and spread influence at the same time. That was the only part of the thesis that does not concern voting, but rather a linear combination over two greedy algorithms. I showed that under some conditions, we can spread influence as well as the classical greedy influence maximization algorithm, but mapping the network much better.

Concerning Team Assessment, I presented a novel technique to predict whether a team of voting agents will be successful or will fail while solving a complex problem. My prediction can be performed online, and it is completely domain independent. I demonstrate its effectiveness in the Computer Go domain, and in the Ensemble Learning domain, and I also showed that the quality of the prediction increases in large action spaces.

Although this thesis presents several contributions, in order to effectively use multi-agent teams to jointly take decisions into solving complex problems much more must be developed. Some topics I will investigate in my future work include:

Dynamic Teams: For effective teamwork, we must dynamically change teams. Hence, we need to further develop assessment mechanisms, and also methods to decide how to switch members and coordination methods according to the assessment results and the problem state.

Agent Adaptation for Teamwork: Although some agents may already exist for a certain problem, most often they were not originally designed to coordinate with others, or were not designed to coordinate using our desired mechanism. Therefore, enabling an easy re-use of agents for teamwork will open the doors for a large number of agents available to choose from.

Team Generation: For many problems we may not even have a few agents available to choose from; in fact, there could be only one agent. My research shows, however, that diversity is an important factor in teamwork. Therefore, given a “basic” agent, automatic methods to create diverse variations of it are very desirable, as a way to easily form diverse teams. As I showed in the thesis, generating random parametrizations of an agent is not yet a good solution; as the resulting team may still have limited diversity, besides the risk of the individual members being too weak in comparison with the “basic” agent.

Mixed Teams: In order to effectively use human input, AI systems must present complex and large-scale information in a manageable way, while simultaneously gaining user trust. Additionally, we must not assume that an “optimal” AI system is always “correct”, since any model has limitations. Hence, the system must also respect and trust the user opinion, delegating when necessary and even changing its own behavior accordingly. Besides, I believe that an interaction between a human and the computer is essential when handling projects related to arts and creativity.

Simultaneous Influencing and Mapping: In Chapter 7, I studied the problem of learning a social network graph, while spreading influence. However, in order to effectively solve this problem, a deeper study is necessary. In particular, it is important to better understand how we can estimate beforehand the amount of information that we will

obtain when interviewing one node (according, for example, to its profile). We must also understand better how the information provided by a node correlates with the list of neighbors in the social network graph. That is, we must understand how likely will be someone to teach us about each one of his/her friends, and how likely would he/she teach us about nodes that are further away in the social network graph; allowing us to better model the knowledge gain at each intervention, and hence improve our performance.

References

- Noa Agmon and Peter Stone. Leading ad hoc agents in joint action settings with multiple teammates. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '12, pages 341–348, Richland, SC, 2012.
- Angela A Aidala and Esther Sumartojo. Why housing? *AIDS and Behavior*, 11(2):1–6, 2007.
- Sandip Aine, Siddharth Swaminathan, Venkatraman Narayanan, Victor Hwang, and Maxim Likhachev. Multi-heuristic A*. In *Proceedings of Robotics: Science and Systems X*, RSS, 2014.
- Abdullah AL-Malaise, Areej Malibari, and Mona Alkhozai. Students' performance prediction system using multi agent data mining technique. *International Journal of Data Mining & Knowledge Management Process*, 4(5), September 2014.
- Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph (Seffi) Naor. The online set cover problem. *SIAM Journal on Computing*, 39(2), 2009.
- Georgios Amanatidis, Nathanaël Barrot, Jérôme Lang, Evangelos Markakis, and Bernard Ries. Multiple referenda and multiwinner elections using hamming distances: Complexity and manipulability. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2015.
- Aris Anagnostopoulos, Diodato Ferraioli, and Stefano Leonardi. Competitive influence in social networks: Convergence, submodularity, and competition effects. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2015.
- Pierpaolo Andriani and Bill McKelvey. Beyond gaussian averages: Redirecting international business and management research toward extreme events and power laws. *Journal of International Business Studies*, 38(7), 2007.
- B. Aranda and C. Lasch. *Flocking, in Tooling*. Princeton Architectural Press, 2006.
- Haris Aziz, Serge Gaspers, Joachim Gudmundsson, Simon Mackenzie, Nicholas Mattei, and Toby Walsh. Computational aspects of multi-winner approval voting. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2015.

Yoram Bachrach, Thore Graepel, Gjergji Kasneci, Michal Kosinski, and Jurgen Van Gael. Crowd IQ: Aggregating opinions to boost performance. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, pages 535–542, 2012.

Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, 2014.

Eyal Baharad, Jacob Goldberger, Moshe Koppel, and Shmuel Nitzan. Distilling the wisdom of crowds: weighted aggregation of decisions on multiple issues. *Journal of Autonomous Agents and Multi-Agent Systems*, 22:31–42, 2011.

E. Baharlou and A. Menges. Generative agent-based design computation, in computation and performance. In *Proceedings of the 31st Education and research in Computer Aided Architectural Design in Europe Conference*, eCAADe, 2013.

Sander Bakkes, Pieter Spronck, and Jaap van den Herik. Opponent modelling for case-based adaptive game AI. *Entertainment Computing*, 1(1):27–37, 2009.

Bikramjit Banerjee and Peter Stone. General game learning using knowledge transfer. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI, 2007.

Samuel Barret and Peter Stone. Cooperating with unknown teammates in complex domains: A robot soccer case study of ad hoc teamwork. In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence*, AAI, 2015.

Adam Barth, Benjamin I. P. Rubinstein, Mukund Sundararajan, John C. Mitchell, Dawn Song, and Peter L. Bartlett. A learning-based approach to reactive security. In Radu Sion, editor, *Financial Cryptography and Data Security – Revised Selected Papers of the Proceedings of the Financial Cryptography and Data Security Conference*, volume 6052 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2010.

J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989a.

J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(3):157–165, 1989b.

Mohammadhossein Bateni, Mohammadtaghi Hajiaghayi, and Morteza Zadimoghaddam. Submodular secretary problem and extensions. *ACM Transactions on Algorithms*, 9(4), 2013.

Petr Baudiš and Jean-loup Gailly. Pachi: State of the Art Open Source Go Program. In *Advances in Computer Games 13*, November 2011.

- Alexander Berman and Valencia James. Kinetic imaginations: Exploring the possibilities of combining AI and dance. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*, 2015.
- D. S. Bernstein, C. Amato, E. Hansen, and S. Zilberstein. Policy iteration for decentralized control of markov decision processes. *Journal of Artificial Intelligence Research*, 34:89–132, 2009.
- Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27:819–840, 2002.
- Dimitri P. Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, 1(1):7–66, 1992.
- Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, and Iain Matthews. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *Proceedings of the 2014 IEEE International Conference on Data Mining, ICDM*, 2014.
- U. Bogenstätter. Prediction and optimization of life-cycle costs in early design. *Building, Research & Information*, 28:376–386, 2000.
- C. Boutilier, I. Caragiannis, S. Haber, T. Lu, A. D. Procaccia, and O. Sheffet. Optimal social choice functions: A utilitarian view. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 197–214, 2012.
- Craig Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of the 18th National Conference on Artificial Intelligence, AAAI*, 2002.
- Florian Brandl, Felix Brandt, and Johannes Hofbauer. Incentives for participation and abstention in probabilistic social choice. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2015.
- Yann Braouezec. Committee, expert advice, and the weighted majority algorithm: An application to the pricing decision of a monopolist. *Computational Economics*, 35(3):245–267, March 2010.
- Nils Bulling, Mehdi Dastani, and Max Knobbout. Monitoring norm violations in multi-agent systems. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2013.
- Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002.
- Adam Burnett, Evon Khor, Philippe Pasquier, and Arne Eigenfeldt. Validation of harmonic progression generator using classical music. In *Proceedings of the Third International Conference on Computational Creativity, ICC3*, 2012.

- Ioannis Caragiannis, Ariel D. Procaccia, and Nisarg Shah. When do noisy votes reveal the truth? In *Proceedings of the 14th ACM conference on Electronic Commerce*, 2013.
- Pablo Miranda Carranza and Paul Coates. Swarm modelling: the use of swarm intelligence to generate architectural form. In *Proceedings of the 3rd International Generative Art Conference*, Generative Art, 2000.
- U. Cekmez, M. Ozsignan, and O. K. Sahingoz. Adapting the GA approach to solve traveling salesman problems on CUDA architecture. In *Proceedings of the IEEE 14th International Symposium on Computational Intelligence and Informatics*, CINTI, 2013.
- U. Cekmez, M. Ozsignan, M. Aydin, and O. K. Sahingoz. UAV path planning with parallel genetic algorithms on CUDA architecture. In *International Conference of Parallel and Distributed Computing*, 2014.
- Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- Shurojit Chatterji, Arunava Sen, and Huaxia Zeng. Random dictatorship domains. *Games and Economic Behavior*, 86:212–236, 2014.
- H. Chen and X. Yao. Regularized negative correlation learning for neural network ensembles. *IEEE Transactions on Neural Networks*, 20(12), 2009.
- Bark Cheung Chiu and Geoffrey I. Webb. Using decision trees for agent modeling: Improving prediction performance. *User Modeling and User-Adapted Interaction*, 8:131–152, 1998.
- Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F. Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. Technical report, Microsoft Research, 2014.
- P. R. Cohen and H. J. Levesque. Confirmation and joint action. In *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI, 1991.
- Marquis de Condorcet. *Essai sur l'application de l'analyse a la probabillite des decisions rendues a la pluralite des voix*. L'Imprimerie Royale, 1785.
- V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI, 2003.
- Vincent Conitzer and Tuomas Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the Twenty-First Conference Conference on Uncertainty in Artificial Intelligence*, UAI, 2005.
- J. Davies, G. Katsirelos, N. Narodytska, and T. Walsh. Complexity of and algorithms for borda manipulation. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, AAAI, 2011.

- Swapnil Dhamal, Prabuchandran K. J., and Yadati Narahari. A multi-phase approach for improving information diffusion in social networks. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2015.
- Mark d’Inverno and Jon McCormack. Heroic versus Collaborative AI for the arts. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*, 2015.
- Thu Trang Doan, Yuan Yao, Natasha Alechina, and Brian Logan. Verifying heterogeneous multi-agent programs. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2014.
- T. M. Echenagucia, A. Capozzoli, Y. Cascone, and M. Sassone. The early design stage of a building envelope: Multi-objective search through heating, cooling and lighting energy performance analysis. *Applied Energy*, 154:577–591, 2015.
- Edith Elkind and Nisarg Shah. Electing the most probable without eliminating the irrational: Voting over intransitive domains. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence, UAI*, 2014.
- M. Enzenberger, M. Müller, B. Arneson, and R. Segal. Fuego - An open-source framework for board games and go engine based on Monte Carlo Tree Search. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(4):259–270, dec. 2010.
- Halil Erhan, Ivy Wang, and Naghmi Shireen. Interacting with thousands: A parametric-space exploration method in generative design. In *Proceedings of the 2014 Conference of the Association for Computer Aided Design in Architecture, ACADIA*, 2014.
- Robert S. Erikson and Christopher Wlezien. Are political markets really superior to polls as election predictors? *The Public Opinion Quarterly*, 72(2):190–215, 2008.
- Joseph Farfel and Vincent Conitzer. Aggregating value ranges: Preference elicitation and truthfulness. *Journal of Autonomous Agents and Multi-Agent Systems*, 22:127–150, 2011.
- J. Felkner, E. Chatzi, and T. Kotnik. Interactive particle swarm optimization for the architectural design of truss structures. In *Proceedings of the 2013 IEEE Symposium in Computational Intelligence for Engineering Solutions, CIES*, pages 15–22, 2013.
- Y. Freund and R.E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29, 1999.
- FSF. Gnugo. <http://www.gnu.org/software/gnugo/>, 2009.

- Bin Fu, Zhihai Wang, Rong Pan, Guandong Xu, and Peter Dolog. An integrated pruning criterion for ensemble learning based on classification accuracy and diversity. In Lorna Uden, Francisco Herrera, Javier Bajo Perez, and Juan M. Corchado Rodriguez, editors, *Proceedings of the 7th International Conference on Knowledge Management in Organizations: Service and Cloud Computing, KMO'12*, volume 172 of *Advances in Intelligent Systems and Computing*, pages 47–58. Springer, 2012.
- Sam Ganzfried and Tuomas Sandholm. Game theory-based opponent modeling in large imperfect-information games. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2011.
- Matthew E. Gaston and Marie desJardins. Agent-organized networks for dynamic team formation. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 230–237, New York, NY, USA, 2005. ACM.
- Sylvain Gelly, Yizao Wang, Rémi Munos, and Olivier Teytaud. Modification of UCT with patterns in Monte-Carlo Go. Technical report, Institut National de Recherche en Informatique et en Automatique, 2006.
- Katie Genter and Peter Stone. Influencing a flock via ad hoc teamwork. In *Proceedings of the Ninth International Conference on Swarm Intelligence, ANTS*, 2014.
- D. J. Gerber and S.-H. E. Lin. Designing in complexity: Simulation, integration, and multidisciplinary design optimization for architecture. *Simulation*, April 2013.
- David J. Gerber. *Parametric practices: Models for design exploration in architecture*. PhD thesis, Harvard University, 2007.
- J. S. Gero. Computational models of innovative and creative design processes. *Technological Forecasting and Social Change*, 64:183–196, 2000.
- J. S. Gero and R. Sosa. Complexity measures as a basis for mass customization of novel designs. *Environment and Planning B: Planning and Design*, 35(1):3–15, 2008.
- Anastasia Globa, Michael Donn, and Jules Moloney. Abstraction versus case-based: A comparative study of two approaches to support parametric design. In *Proceedings of the 2014 Conference of the Association for Computer Aided Design in Architecture, ACADIA*, 2014.
- Daniel Golovin and Andreas Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. In *Proceedings of the 23rd International Conference on Learning Theory*, 2010.
- Carla P. Gomes and Bart Selman. Algorithm portfolios. *Artificial Intelligence*, 126: 43–62, 2001.

- F. Grandoni, A. Gupta, S. Leonardi, P. Miettinen, P. Sankowski, and M. Singh. Set covering with our eyes closed. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 2008.
- B. Grosz and S. Kraus. Collaborative plans for complex group actions. *Artificial Intelligence*, 86:269–358, 1996.
- Christian Guttman. Making allocations collectively: Iterative group decision making under uncertainty. In Ralph Bergmann, Gabriela Lindemann, Stefan Kirn, and Michal Pechoucek, editors, *Proceedings of the 6th German Conference on Multi-agent System Technologies*, volume 5244 of *Lecture Notes in Computer Science*, pages 73–85, Kaiserslautern, Germany, 2008. Springer.
- Andras Gyorgy and Levente Kocsis. Efficient multi-start strategies for local search algorithms. *Journal of Artificial Intelligence Research*, 41:407–444, 2011.
- David J. Hand and Robert J. Till. A simple generalization of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45:171–186, 2001.
- James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.
- Sean Hanna. Where creativity comes from? The social spaces of embodied minds. In *Proceedings of the International Conference of Computational and Cognitive Models of Creative Design VI*, 2005.
- Graeme A. Haynes. Testing the boundaries of the choice overload phenomenon: The effect of number of options and time pressure on decision difficulty and satisfaction. *Psychology & Marketing*, 26(3):204–212, 2009.
- Linli He and Thomas R. Ioerger. A quantitative model of capabilities in multi-agent systems. In Hamid R. Arabnia, Rose Joshua, and Youngsong Mun, editors, *Proceedings of the International Conference on Artificial Intelligence, IC-AI*, pages 730–736, Las Vegas, USA, 2003. ISBN 1-932415-13-0.
- Erik L Heiny and David Blevins. Predicting the atlanta falcons play-calling using discriminant analysis. *Journal of Quantitative Analysis in Sports*, 7(3), July 2011.
- Ernst Hellinger. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 136:210–271, 1909.
- D. Holzer, R. Hough, and M. Burry. Parametric design and structural optimisation for early design exploration. *International Journal of Architectural Computing*, 5: 625–643, 2007.
- Lu Hong and Scott E. Page. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences of the USA*, 101(46):16385–16389, 2004.

- Josie Hunter, Franco Raimondi, Neha Rungta, and Richard Stocker. A synergistic and extensible framework for multi-agent system verification. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2013.
- T. Ireland. Emergent space diagrams: The application of swarm intelligence to the problem of automatic plan generation. In *Proceedings of the 13th International CAAD Futures Conference*, 2009.
- I. S. Isa, S. Omar, Z. Saad, N. M. Noor, and M. K. Osman. Weather forecasting using photovoltaic system and neural network. In *2010 Second International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN*, pages 96–100, July 2010.
- S. Iyengar and M. Lepper. When choice is demotivating: Can one desire too much of a good thing? *Journal of Personality and Social Psychology*, 79:995–1006, 2000.
- A. X. Jiang, L. S. Marcolino, A. D. Procaccia, T. Sandholm, N. Shah, and M. Tambe. Diverse randomized agents vote to win. In *Proceedings of the Neural Information Processing Systems Conference, NIPS*, 2014.
- Meir Kalech and Gal A. Kaminka. On the design of coordination diagnosis algorithms for teams of situated agents. *Artificial Intelligence*, 171:491–513, 2007.
- Meir Kalech, Sarit Kraus, Gal A. Kaminka, and Claudia V. Goldman. Practical voting rules with partial information. *Journal of Autonomous Agents and Multi-Agent Systems*, 22:151–182, 2011.
- Gal A. Kaminka. *Coordination of Large-Scale Multiagent Systems*, chapter Handling Coordination Failures in Large-Scale Multi-Agent Systems. Springer, 2006.
- Gal A. Kaminka and Milind Tambe. What is wrong with us? Improving robustness through social diagnosis. In *Proceedings of the National Conference on Artificial Intelligence, AAI*, 1998.
- Jeffrey A. Kelly, Debra A. Murphy, Kathleen J. Sikkema, Timothy L. McAuliffe, Roger A. Roffman, Laura J. Solomon, Richard A. Winett, and Seth C. Kalichman. Randomised, controlled, community-level HIV-prevention intervention for sexual-risk behaviour among homosexual men in US cities. *The Lancet*, 350(9090):1500, 1997.
- David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, 2003.
- Ian Keough and David Benjamin. Multi-objective optimization in architectural design. In *Proceedings of the 2010 Spring Simulation Multiconference, SpringSim*, 2010.

- Eliahu Khalastchi, Meir Kalech, and Lior Rokach. A hybrid approach for fault detection in autonomous physical agents. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2014.
- Samir Khuller, Anna Moss, and Joseph (Seffi) Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1), 1999.
- D. S. Knysh and V. M. Kureichik. Parallel genetic algorithms: A survey and problem state of the art. *Journal of Computer and Systems Sciences International*, 49(4):579–589, 2010.
- George Konidaris, Ilya Scheidwasser, and Andrew G Barto. Transfer in reinforcement learning via shared features. *The Journal of Machine Learning Research*, 13(1), 2012.
- Panagiotis Kouvaros and Alessio Lomuscio. Automatic verification of parameterised interleaved multi-agent systems. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2013.
- Stefan Krause, Richard James, Jolyon J. Faria, Graeme D. Ruxton, and Jens Krause. Swarm intelligence in humans: diversity can trump ability. *Animal Behaviour*, 81(5):941–948, May 2011.
- Karim R. Lakhani, Lars Bo Jeppesen, Peter A. Lohse, and Jill A. Panetta. The value of openness in scientific problem solving. *HBS Working Paper*, 07-050, 2007. URL <http://hbswk.hbs.edu/item/5612.html>.
- P. J. Lamberson and Scott E. Page. Optimal forecasting groups. *Management Science*, 58(4):805–810, 2012.
- Zhaofeng Li and Yichuan Jiang. Cross-layers cascade in multiplex networks. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2014.
- Marco LiCalzi and Oktay Surucu. The power of diversity over large solution spaces. *Management Science*, 58(7):1408–1421, July 2012. URL <http://dx.doi.org/10.1287/mnsc.1110.1495>.
- Somchaya Liemhetcharat and Manuela Veloso. Modeling and learning synergy for team formation with heterogeneous agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, pages 365–374, Richland, SC, 2012.
- Shih-hsin Eve Lin and David Jason Gerber. Designing-in performance: A framework for evolutionary energy performance feedback in early stage design. *Automation and Construction*, 38:59–73, 2012.
- Shih-Hsin Eve Lin and David Jason Gerber. Evolutionary energy performance feedback for design. *Energy and Buildings*, 84:426–441, 2014.

- Michael Q. Lindner and Noa Agmon. Effective, quantitative, obscured observation-based fault detection in multi-agent systems. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2014.
- Christian List and Robert E. Goodin. Epistemic democracy: Generalizing the Condorcet Jury Theorem. *Journal of Political Philosophy*, 9:277–306, 2001.
- Alan J. Lockett, Charles L. Chen, and Risto Miikkulainen. Evolving explicit opponent models in game playing. In *Proceedings of the 9th annual conference on genetic and evolutionary computation*, GECCO, 2007.
- Shenghua Luan, Konstantinos V. Katsikopoulos, and Torsten Reimer. When does diversity trump ability (and vice versa) in group decision making? A simulation study. *PLoS One*, 7(2):e31043, 2012.
- Patrick Lucey, Alina Bialkowski, Peter Carr, Yisong Yue, and Iain Matthews. How to get an open shot: Analyzing team movement in basketball using tracking data. In *Proceedings of the 8th Annual MIT SLOAN Sports Analytics Conference*, 2014.
- Patrick Lucey, Alina Bialkowski, Mathew Monfort, Peter Carr, and Iain Matthews. Quality vs quantity: Improved shot prediction in soccer using strategic features from spatiotemporal data. In *Proceedings of the 9th Annual MIT SLOAN Sports Analytics Conference*, 2015.
- Guan-Chun Luh and Chun-Yi Lin. Structural topology optimization using ant colony optimization algorithm. *Applied Soft Computing*, 9:1343–1353, 2009.
- Guan-Chun Luh, Chun-Yi Lin, and Yu-Shu Lin. A binary particle swarm optimization for continuum structural topology optimization. *Applied Soft Computing*, 11: 2833–2844, 2011.
- Penousal Machado, Adriano Vinhas, João Correia, and Aniko Ekárt. Evolving ambiguous images. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI, 2015.
- Mahsa Maghami and Gita Sukthankar. Identifying influential agents for advertising in multi-agent markets. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2012.
- R. Maheswaran, Y. Chang, A. Henehan, and S. Danesis. Deconstructing the rebound with optical tracking data. In *Proceedings of the 6th Annual MIT SLOAN Sports Analytics Conference*, 2012.
- Charles F. Manski. Interpreting the predictions of prediction markets. *Economics Letters*, 91(3):425–429, June 2006.
- Andrew Mao, Ariel D. Procaccia, and Yiling Chen. Better Human Computation Through Principled Voting. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI, 2013.

- L. S. Marcolino, B. Kolev, S. Price, S. P. Veetil, D. Gerber, J. Musil, and M. Tambe. Aggregating opinions to design energy-efficient buildings. In *8th Multidisciplinary Workshop on Advances in Preference Handling*, M-PREF, 2014a.
- L. S. Marcolino, H. Xu, D. Gerber, B. Kolev, S. Price, E. Pantazis, and M. Tambe. Multi-agent team formation for design problems. In *Coordination, Organizations, Institutions and Norms in Agent Systems XI*. Springer-Verlag Lecture Notes in AI, 2016a. (To Appear).
- Leandro Soriano Marcolino and Luiz Chaimowicz. No robot left behind: Coordination to overcome local minima in swarm navigation. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, ICRA, 2008.
- Leandro Soriano Marcolino and Luiz Chaimowicz. Traffic control for a swarm of robots: Avoiding target congestion. In *Proceedings of the 2009 IEEE International Conference on Intelligent Robots and Systems*, IROS, 2009a.
- Leandro Soriano Marcolino and Luiz Chaimowicz. Traffic control for a swarm of robots: Avoiding group conflicts. In *Proceedings of the 2009 IEEE International Conference on Intelligent Robots and Systems*, IROS, 2009b.
- Leandro Soriano Marcolino, Albert Xin Jiang, and Milind Tambe. Multi-agent team formation: Diversity beats strength? In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI, 2013.
- Leandro Soriano Marcolino, Haifeng Xu, Albert Xin Jiang, Milind Tambe, and Emma Bowring. Give a hard problem to a diverse team: Exploring large action spaces. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI, 2014b.
- Leandro Soriano Marcolino, Aravind Lakshminarayanan, Vaishnavh Nagarajan, and Milind Tambe. Every team deserves a second chance: An extended study on predicting team performance. *Journal of Autonomous Agents and Multi-Agent Systems*, 2016b. (Under Review).
- Leandro Soriano Marcolino, Aravind Lakshminarayanan, Amulya Yadav, and Milind Tambe. Simultaneous influencing and mapping social networks (short paper). In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2016c.
- P. V. Marsden. Recent developments in network measurement. In Peter J. Carrington, John Scott, and Stanley Wasserman, editors, *Models and methods in social network analysis*. Cambridge University Press, 2005.
- Simon J. Mason and Nicholas E. Graham. Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society*, 128:2145–2166, 2002.

Tim Matthews, Sarvapali D. Ramchurn, and Georgios Chalkiadakis. Competing with humans at Fantasy Football: Team formation in large partially-observable domains. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI, pages 1394–1400, 2012.

Richard D. McKelvey and Thomas R. Palfrey. Quantal response equilibria for normal form games. In *Games and Economic Behavior*, volume 10, pages 6–38. Elsevier, 1995.

J. C. Miles, G. M. Sisk, and C. J. Moore. The conceptual design of commercial buildings using a genetic algorithm. *Computers & Structures*, 79(17):1583–1592, 2001.

Violeta Mirchevska, Mitja Luštrek, Andraž Bežek, and Matjaž Gams. Discovering strategic behaviour of multi-agent systems in adversary settings. *Computing and Informatics*, 2014.

Vaishnavh Nagarajan, Leandro Soriano Marcolino, and Milind Tambe. Every team deserves a second chance: Identifying when things go wrong. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2015. (Double 1st author).

Ranjit Nair and Milind Tambe. Hybrid BDI-POMDP framework for multiagent teaming. *Journal of Artificial Intelligence Research*, 23(1):367–420, April 2005.

S. Nolfi and D. Floreano. *Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-organizing Machines*. MIT Press, Cambridge, MA, 2001.

H. Nurmi. *Comparing Voting Systems*. Springer, 1987.

Takuya Obata, Takuya Sugiyama, Kunihito Hoki, and Takeshi Ito. Consultation algorithm for Computer Shogi: Move decisions by majority. In *Computer and Games'10*, volume 6515 of *Lecture Notes in Computer Science*, pages 156–165. Springer, 2011.

Joel Oren and Brendan Lucier. Online (budgeted) social choice. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, AAAI, 2014.

E. Osaba, E. Onieva, F. Diaz, R. Carballedo, P. Lopez, and A. Perallos. A migration strategy for distributed evolutionary algorithms based on stopping non-promising subpopulations: A case study on routing problems. *International Journal of Artificial Intelligence*, 2015. In Press.

Pandanet. Introduction to Go. http://www.pandanet.co.jp/English/introduction_of_go/, 2016.

Rama Kumar Pasumarthi, Ramasuri Narayanam, and Balaraman Ravindran. Near optimal strategies for targeted marketing in social networks. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2015.

- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Robi Polikar. *Ensemble Machine Learning: Methods and Applications*, chapter Ensemble Learning. Springer, 2012.
- Ariel D. Procaccia, Sashank J. Reddi, and Nisarg Shah. A maximum likelihood approach for selecting sets of alternatives. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, UAI, 2012.
- Jared Quenzel and Paul Shea. Predicting the winner of tied NFL games: Do the details matter? *Journal of Sports Economics*, 2014. Forthcoming, available online.
- A. D. Radford and J. S. Gero. Tradeoff diagrams for the integrated design of the physical environment in buildings. *Building and Environment*, 15(1), 1980.
- T. Raines, M. Tambe, and S. Marsella. Automated assistants to aid humans in understanding team behaviors. In *AGENTS*, 2000.
- Fernando Ramos and Huberto Ayanegui. Discovering tactical behavior patterns supported by topological structures in soccer-agent domains. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2008.
- Paul Rhode and Koleman Strumpf. Historical presidential betting markets. *Journal of Economic Perspectives*, 18(2):127–142, 2004.
- Eric Rice. The positive role of social networks and social networking technology in the condom-using behaviors of homeless young people. *Public health reports*, 125(4):588, 2010.
- Eric Rice and Harmony Rhoades. How should network-based prevention for homeless youth be implemented? *Addiction*, 108(9):1625, 2013.
- Eric Rice, Eve Tulbert, Julie Cederbaum, Anamika Barman Adhikari, and Norweeta G Milburn. Mobilizing homeless youth for HIV prevention: a social network analysis of the acceptability of a face-to-face and online social networking intervention. *Health education research*, 27(2):226, 2012.
- Horst W. J. Rittel and Melvin M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4:155–169, 1973.
- Diederik Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48, 2013.
- F. Schadd, S. Bakkes, and P. Spronck. Opponent modeling in real-time strategy games. In *Proceedings of the 2007 Simulation and AI in Games Conference*, GAMEON, pages 61–68, 2007.

- C. Seshadhri, Tamara G. Kolda, and Ali Pinar. Community structure and scale-free collections of Erdős-Rényi graphs. *Physical Review E*, 85(5):056109, 2012.
- David Silver and Joel Veness. Monte-Carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems*, NIPS, pages 2164–2172, 2010.
- Herbert A. Simon. The structure of ill-structured problems. *Artificial Intelligence*, 4:181–201, 1973.
- Brittany N. Smith, Anbang Xu, and Brian P. Bailey. Improving interaction models for generating and managing alternative ideas during early design work. In *Proceedings of the Graphics Interface Conference*, 2010.
- Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12), December 1980.
- Roland Snooks. Encoding behavioral matter. In *Proceedings of the International Symposium on Algorithmic Design for Architecture and Urban Design*, ALGODE, 2011.
- Yusuke Soejima, Akihiro Kishimoto, and Osamu Watanabe. Evaluating root parallelization in Go. *IEEE Transactions in Computational Intelligence and AI in Games*, 2(4):278 – 287, 2010.
- Hossein Azari Soufiani, David C. Parkes, and Lirong Xia. Random utility theory for social choice. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Proceedings of the Neural Information Processing Systems Conference*, NIPS, pages 126–134, 2012.
- Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes’ bluff: Opponent modeling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI, 2005.
- Peter Stone, Gal A. Kaminka, Sarit Kraus, Jeffrey S. Rosenschein, and Noa Agmon. Teaching and leading an ad hoc teammate: Collaboration without pre-coordination. *Artificial Intelligence Journal*, 203:35–65, October 2013.
- Jared Sylvester and Nitesh V. Chawla. Evolutionary ensembles: Combining learning agents using genetic algorithms. In *Proceedings of the 20th National Conference on Artificial Intelligence*, AAAI, 2005.
- Milind Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.
- Milind Tambe, Paul Scerri, and David V. Pynadath. Adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17(1):171–228, 2002.

- Danesh Tarapore, Anders Lyhne Christensen, Pedro U. Lima, and Jorge Carneiro. Abnormality detection in multiagent systems inspired by the adaptive immune system. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2013.
- Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *The Journal of Machine Learning Research*, 10, 2009.
- Kostas Terzidis. *Algorithmic Architecture*. Routledge, 2006.
- Abigail Thompson. Does diversity trump ability? An example of the misuse of mathematics in the social sciences. *Notices of the AMS*, 61(9), 2014.
- Alan Tsang and Kate Larson. Opinion dynamics of skeptical agents. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2014.
- Hamed Valizadegan, Rong Jin, and Shijun Wang. Learning to trade off between exploration and exploitation in multiclass bandit prediction. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011.
- P.H.G. van Langen and F.M.T. Brazier. Design space exploration revisited. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 20:113–119, 2006.
- Sebastian Vehlken. Computational swarming: A cultural technique for generative architecture. *Footprint – Delft Architecture Theory Journal*, 15, 2014.
- Robert Vierlinger and Klaus Bollinger. Accommodating change in parametric design. In *Proceedings of the 2014 International Conference of the Association for Computer-Aided Design in Architecture*, ACADIA, 2014.
- David West and Scott Dellana. Diversity of ability and cognitive style for group decision processes. *Information Sciences*, 179(5):542–558, 2009.
- Robert F. Woodbury and Andrew L. Burrow. Whither design space? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 20:63–82, 2006.
- Lirong Xia. *Computational voting theory: game-theoretic and combinatorial aspects*. PhD thesis, Duke University, Durham, NC, USA, 2011.
- Lirong Xia and Vincent Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research (JAIR)*, 41, 2011a.
- Lirong Xia and Vincent Conitzer. A maximum likelihood approach towards aggregating partial orders. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, IJCAI, 2011b.

A. Yadav, L. S. Marcolino, E. Rice, R. Petering, H. Winetrobe, H. Rhoades, M. Tambe, and H. Carmichael. Preventing HIV spread in homeless populations using PSINET. In *Proceedings of the Twenty-Seventh Innovative Applications of Artificial Intelligence Conference, IAAI*, 2015.

Yongjie Yang. Manipulation with bounded single-peaked width: A parameterized study. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2015.

Y. K. Yi and A. M. Malkawi. Optimizing building form for energy performance based on hierarchical geometry relation. *Automation in Construction*, 18, 2009.

Peyton Young. Optimal voting rules. *Journal of Economic Perspectives*, 9(1): 51–64, 1995.

Fengqiang Zhao, Guangqiang Li, Chao Yang, Ajith Abraham, and Hongbo Liu. A human–computer cooperative particle swarm optimization based immune algorithm for layout design. *Neurocomputing*, 132:68–78, 2014.

Appendices

Appendix A

SIMULTANEOUS INFLUENCING AND MAPPING: ADDITIONAL RESULTS

A.1 Results for each network

Since in Section 7.3 I presented my results across 4 different social network graphs, in this section I present those individually for each network, for the interested reader. Assuming a uniform model for the teaching lists, we can see the results for network A in Figures A.1, A.2, A.3; for network B in Figures A.4, A.5, A.6; for the Facebook network in Figures A.7, A.8, A.9; and finally for the MySpace network in Figures A.10, A.11, A.12. Assuming a power law model, we can see the results for network A in Figures A.13, A.14, A.15; for network B in Figures A.16, A.17, A.18; for the Facebook network in Figures A.19, A.20, A.21; and finally for the MySpace network in Figures A.22, A.23, A.24. As we can see, the results for each network show similar tendencies as the results across all networks presented in the thesis.

A.2 Additional results for power law distribution

In Section 7.3.2 I presented results for the power law distribution for $a = 1.8$. In this section, I present results for $a = 1.2$ (that is, each neighbor has a 20% probability of being in a teaching list). I show in Figure A.25 the result at each intervention for $\varphi = 0.5$ and $p = 0.5$. In Figure A.26, I show the AUC results for different parametrizations of p and φ . Finally, I present the regret for this case in Figure A.27. As we can see, my main conclusions still hold for a different parametrization of a .

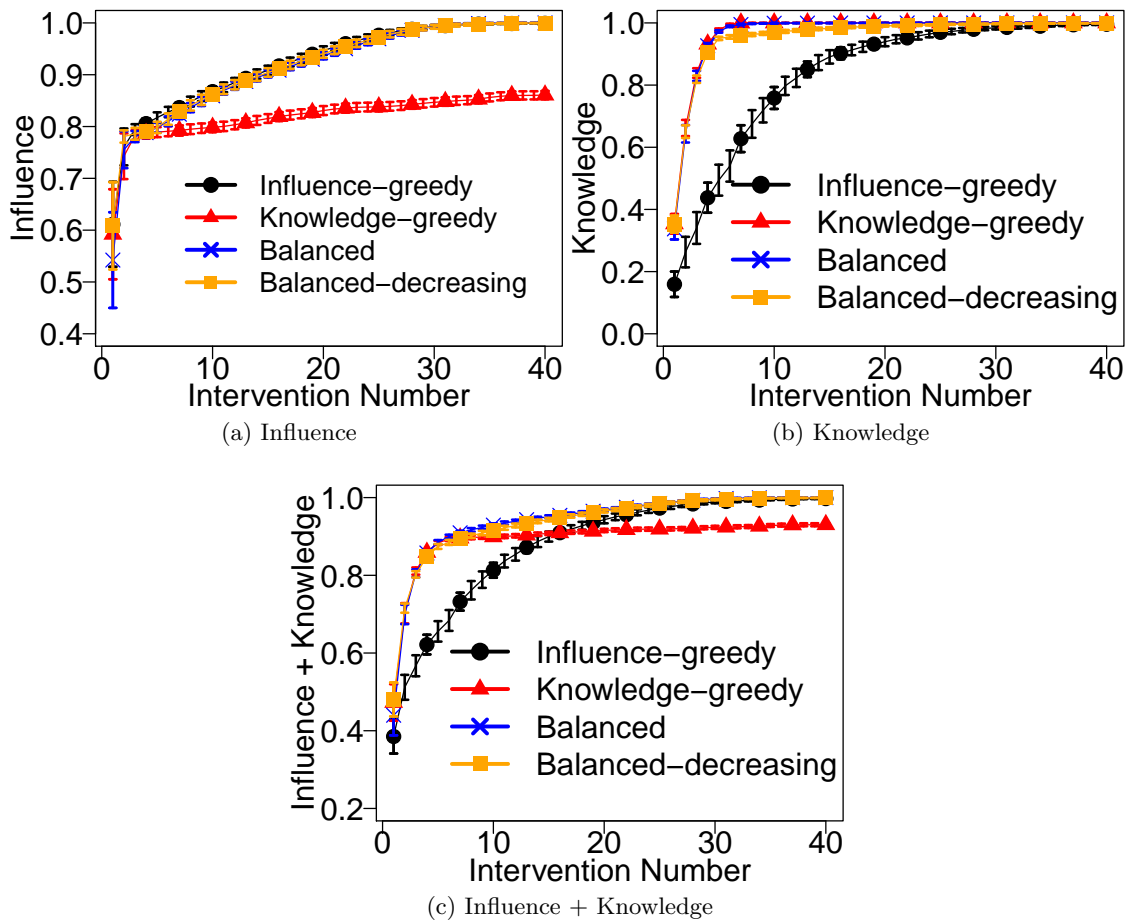
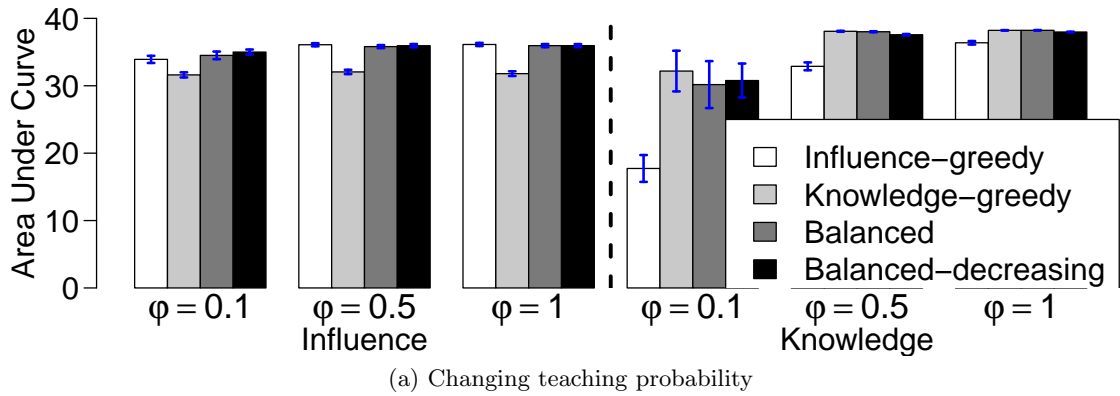
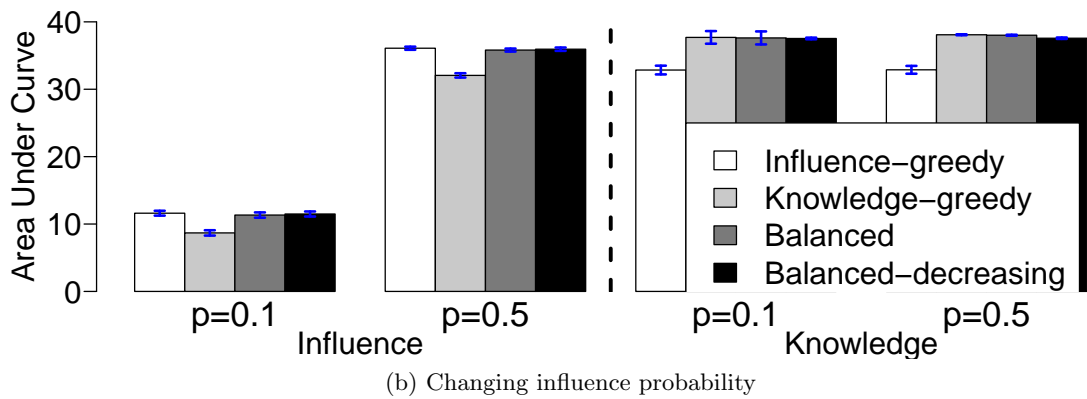


Figure A.1: Results for network A across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming uniform distribution.



(a) Changing teaching probability



(b) Changing influence probability

Figure A.2: Results of Influence and Knowledge in network A for different teaching and influence probabilities, assuming uniform distribution.

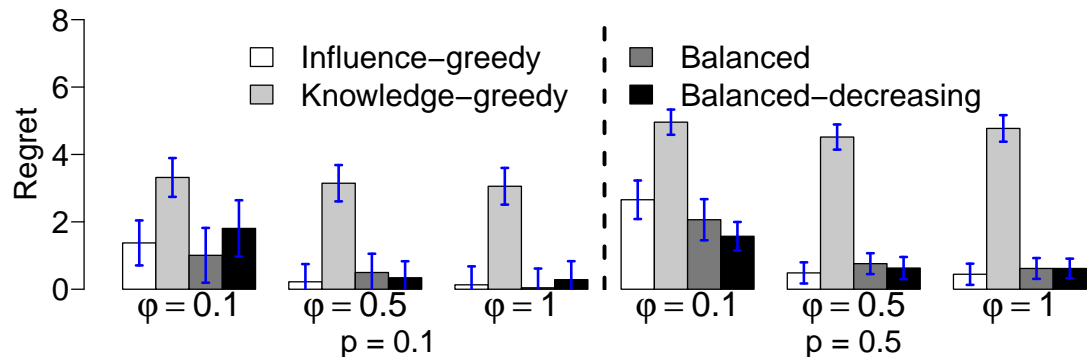


Figure A.3: Regret in network A for different teaching and influence probabilities, assuming uniform distribution.

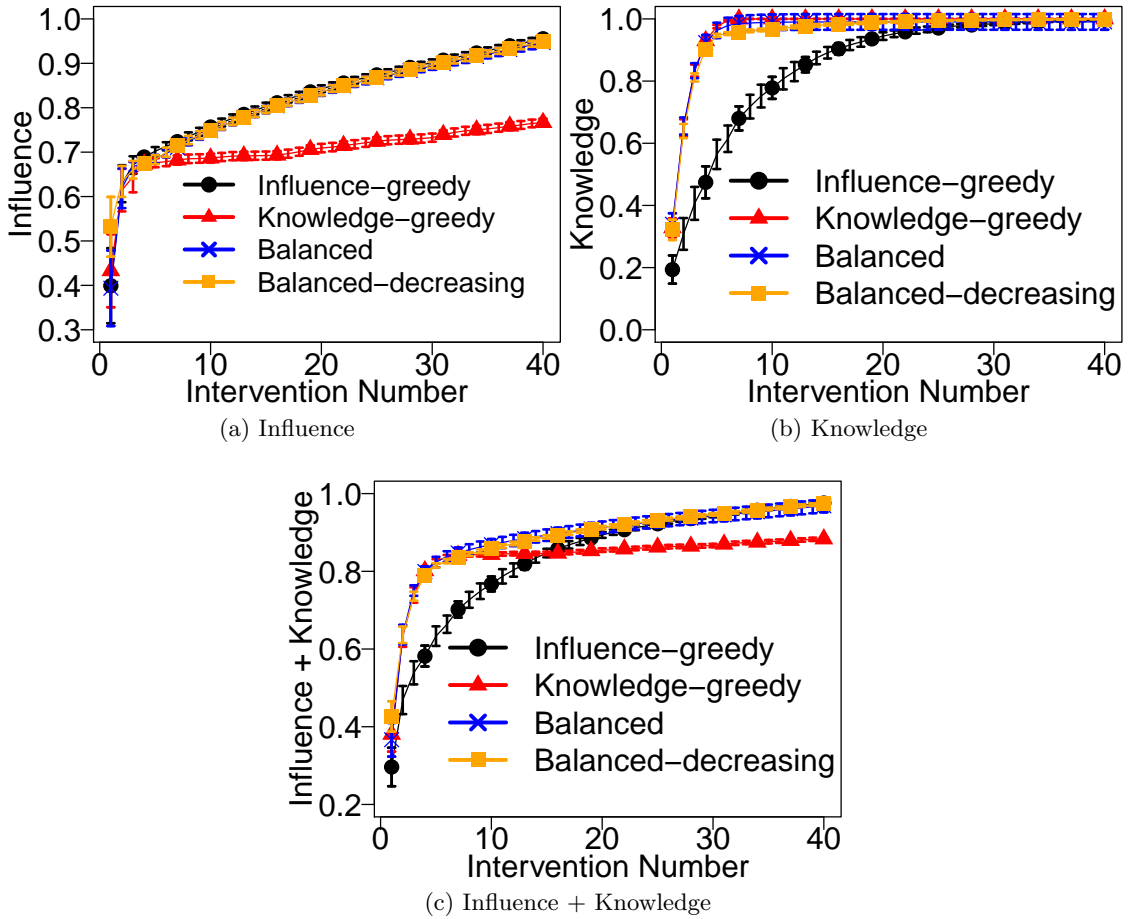


Figure A.4: Results for network B across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming uniform distribution.

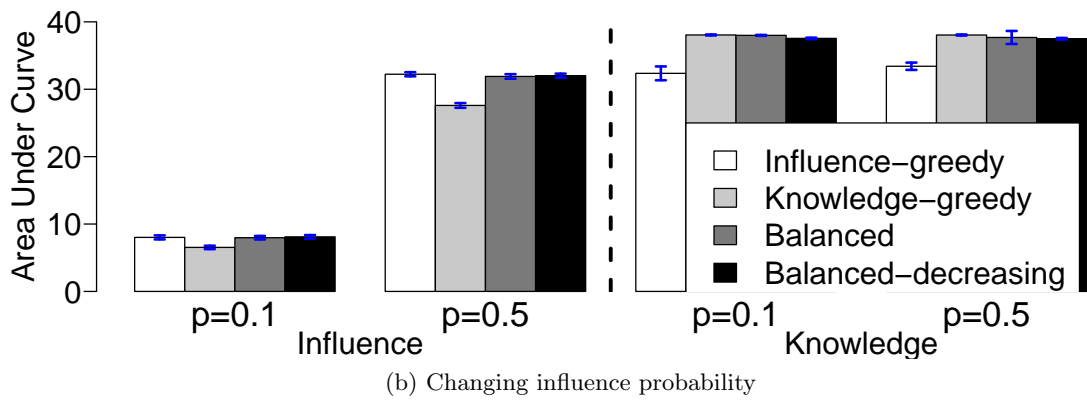
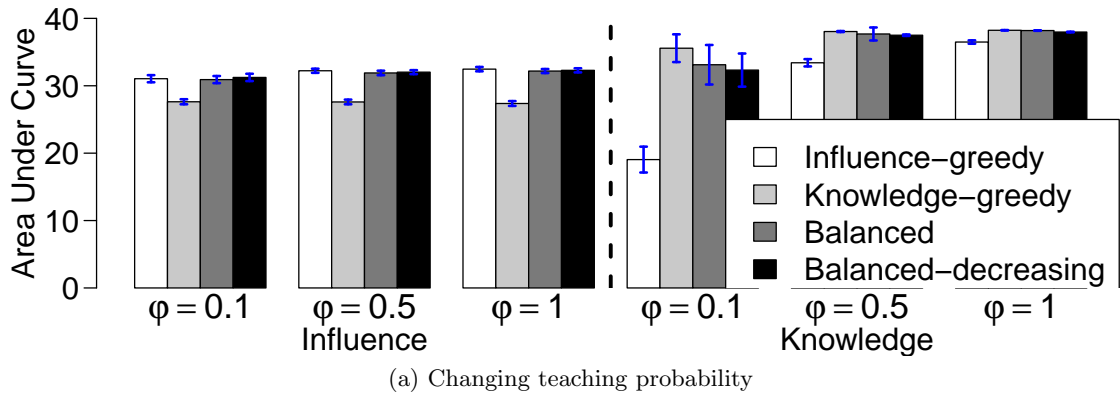


Figure A.5: Results of Influence and Knowledge in network B for different teaching and influence probabilities, assuming uniform distribution.

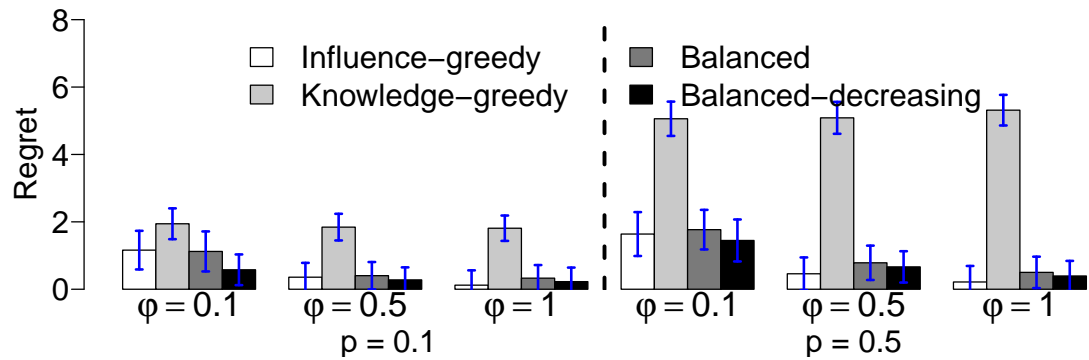


Figure A.6: Regret in network B for different teaching and influence probabilities, assuming uniform distribution.

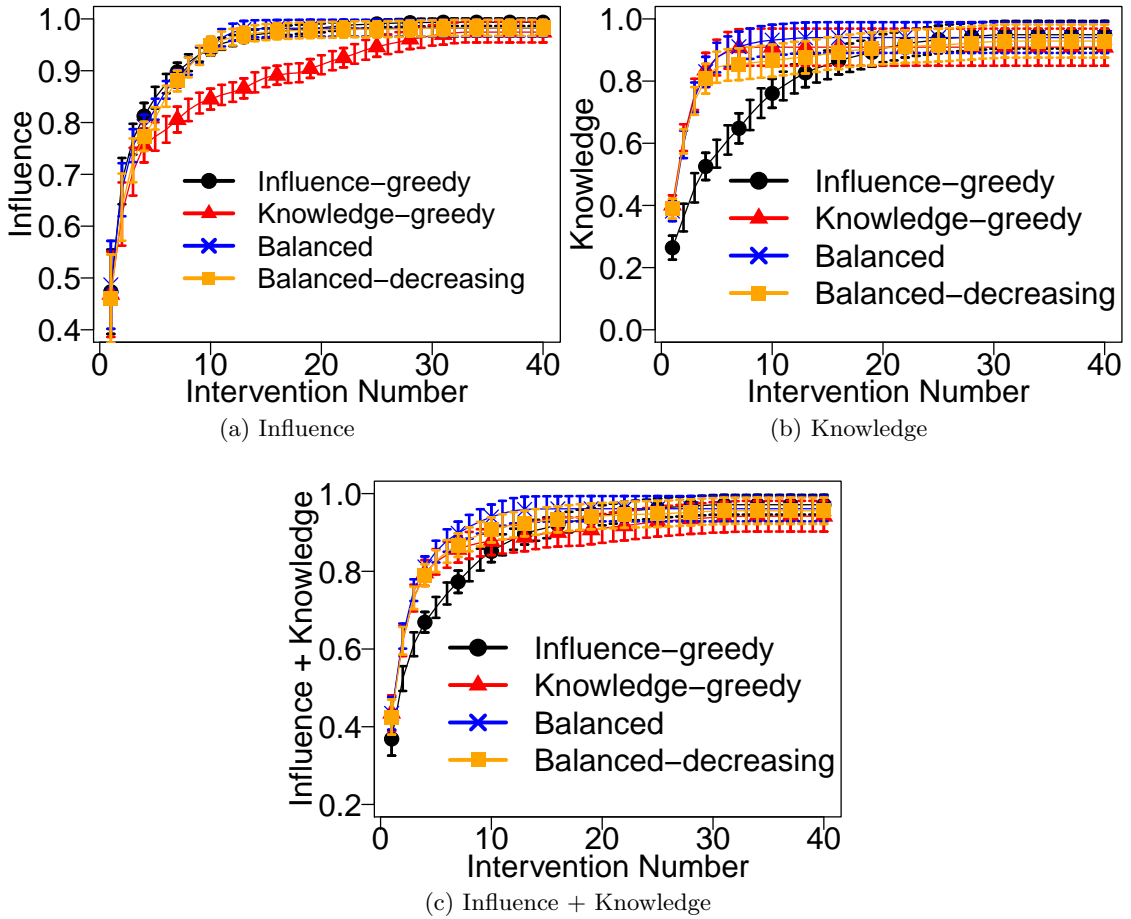


Figure A.7: Results for Facebook network across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming uniform distribution.

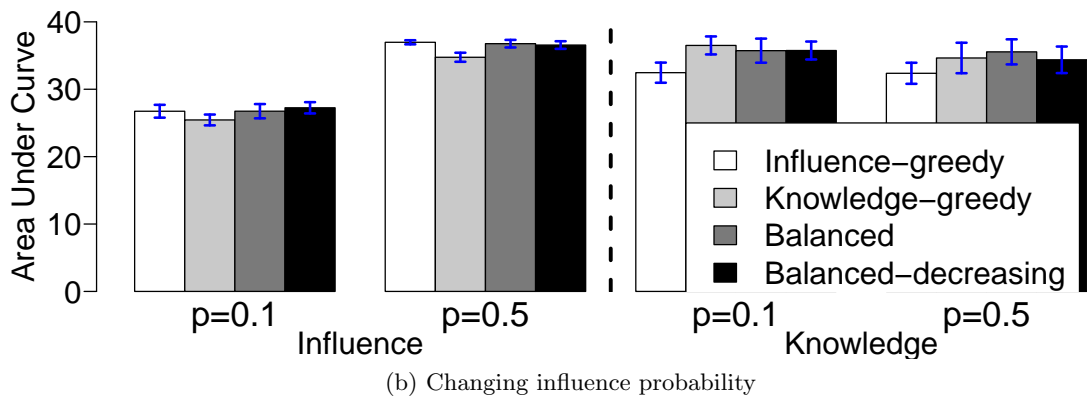
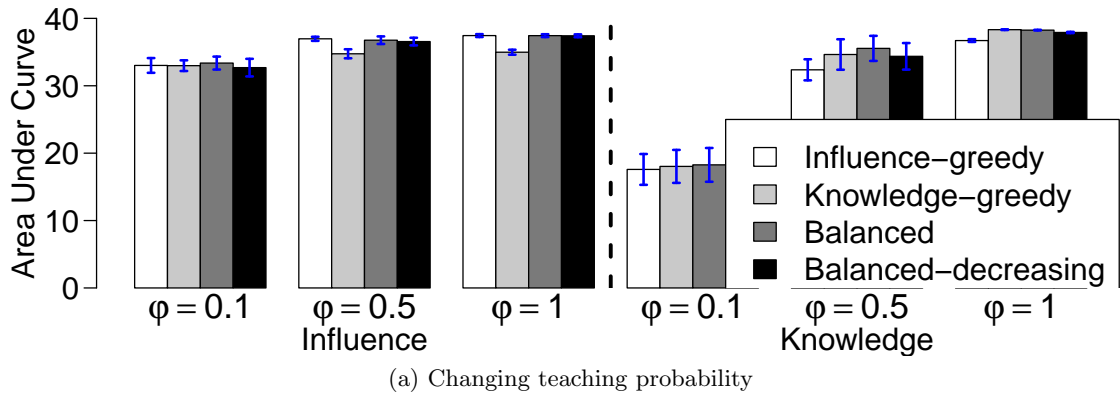


Figure A.8: Results of Influence and Knowledge in Facebook network for different teaching and influence probabilities, assuming uniform distribution.

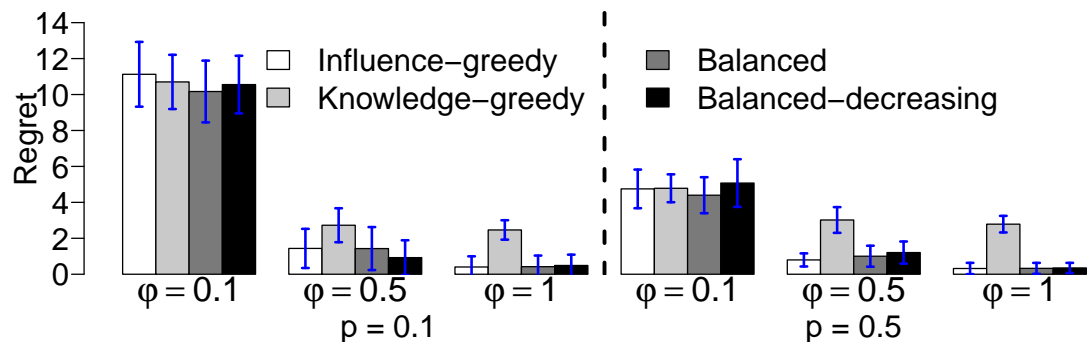


Figure A.9: Regret in Facebook network for different teaching and influence probabilities, assuming uniform distribution.

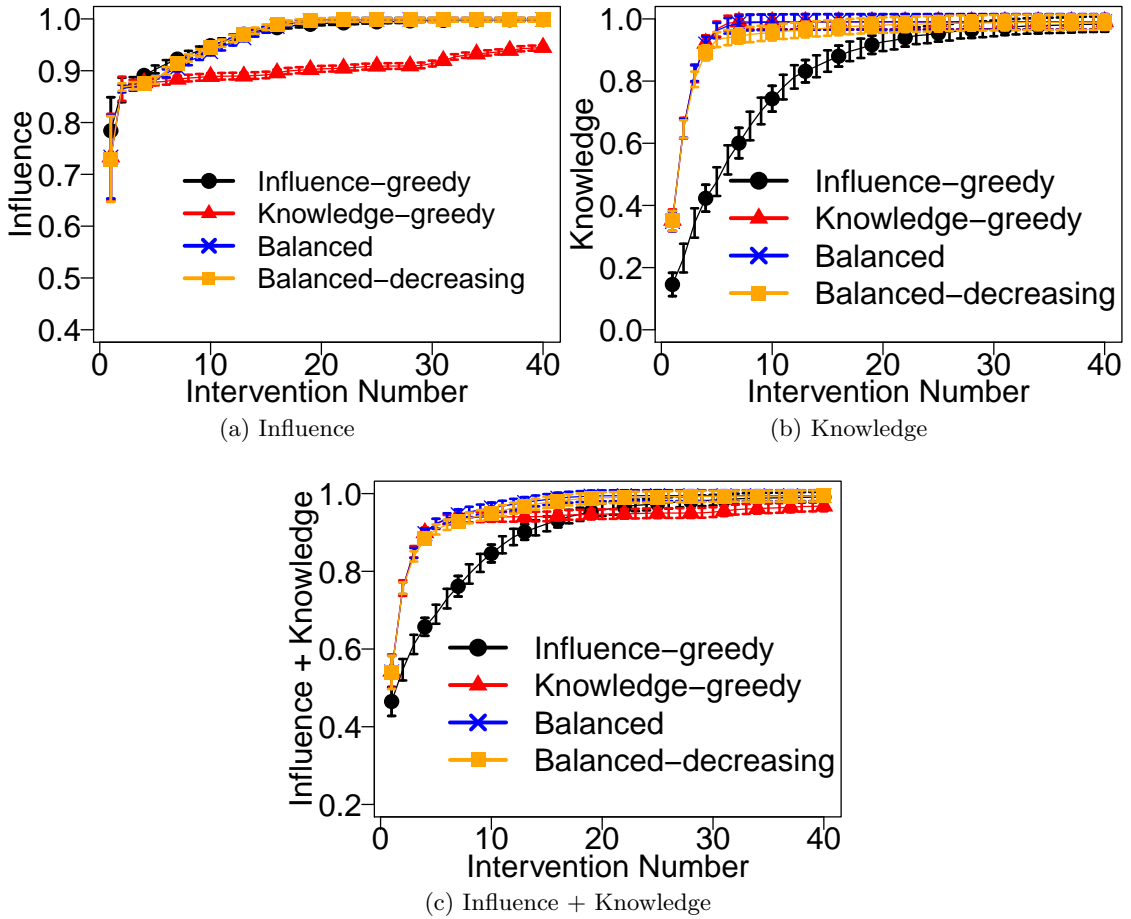
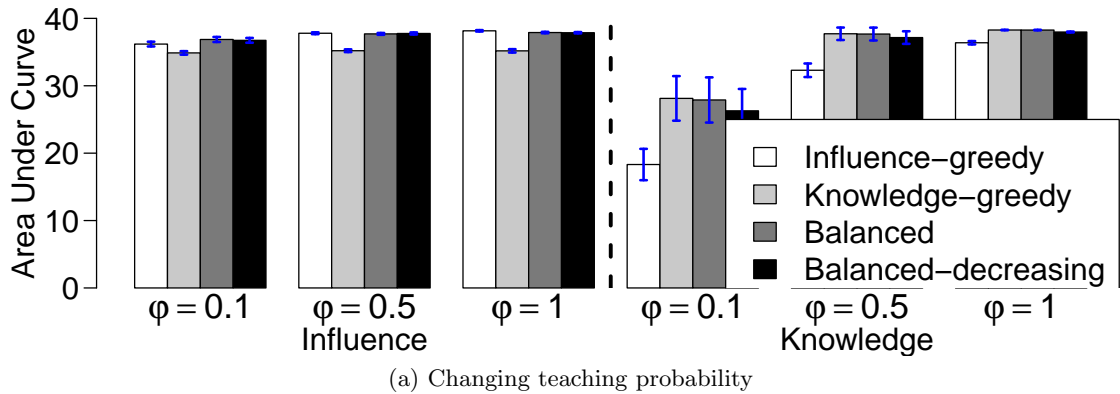
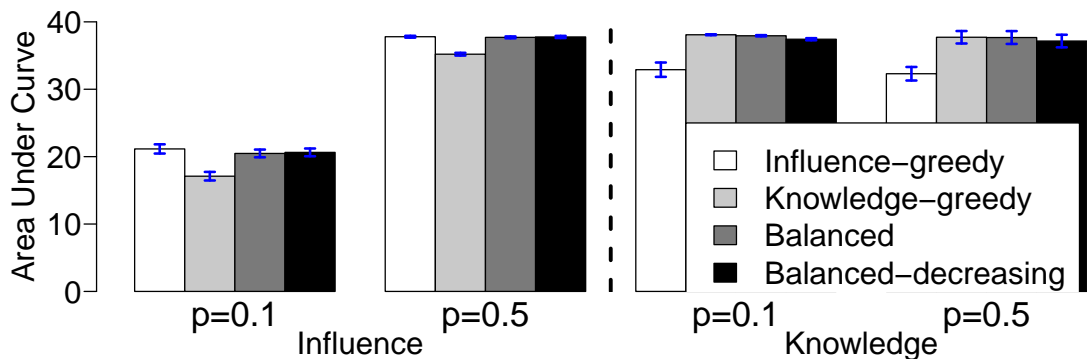


Figure A.10: Results for MySpace network across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming uniform distribution.



(a) Changing teaching probability



(b) Changing influence probability

Figure A.11: Results of Influence and Knowledge in MySpace network for different teaching and influence probabilities, assuming uniform distribution.

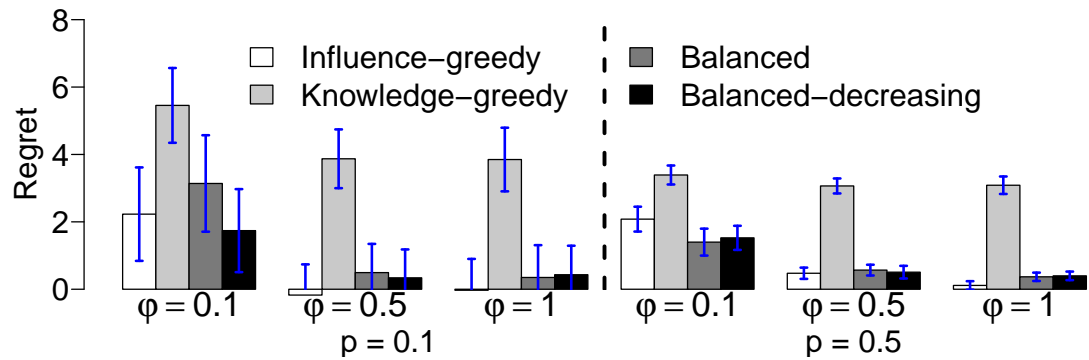


Figure A.12: Regret in MySpace network for different teaching and influence probabilities, assuming uniform distribution.

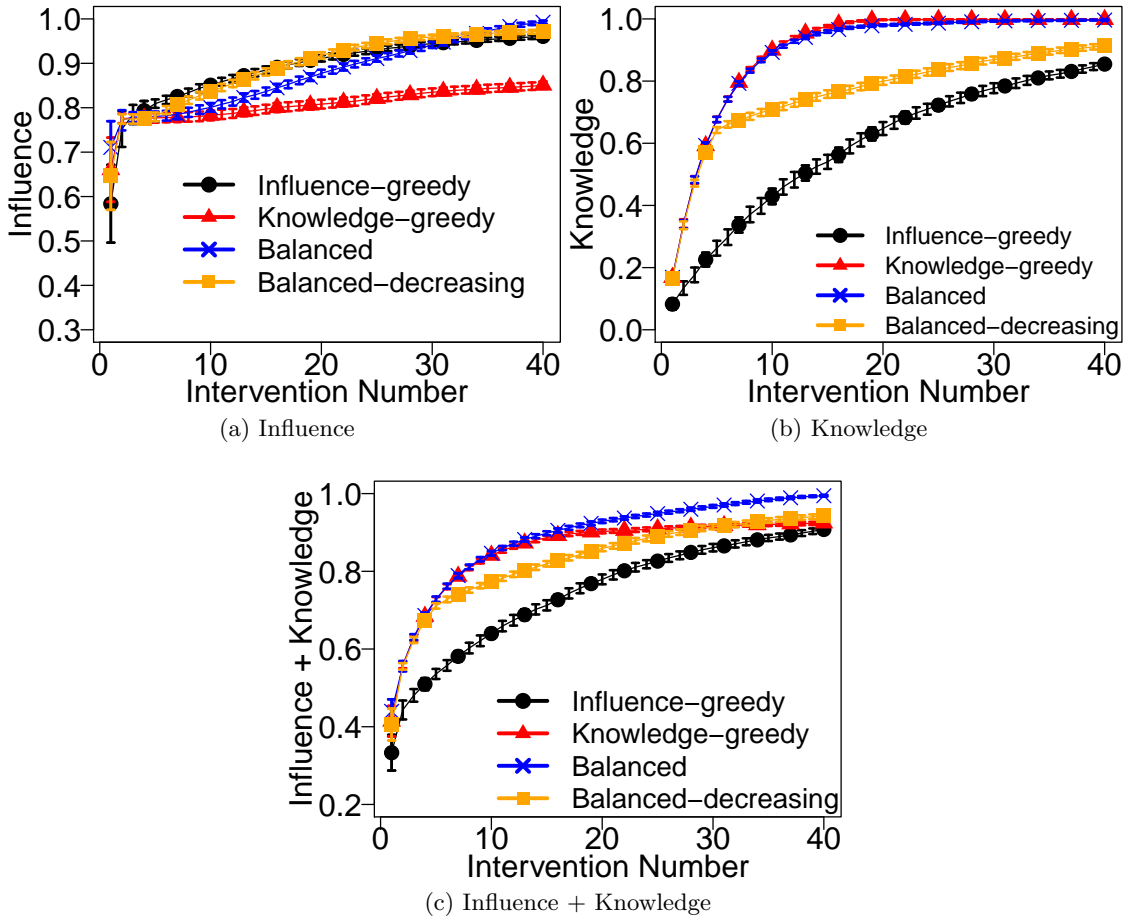
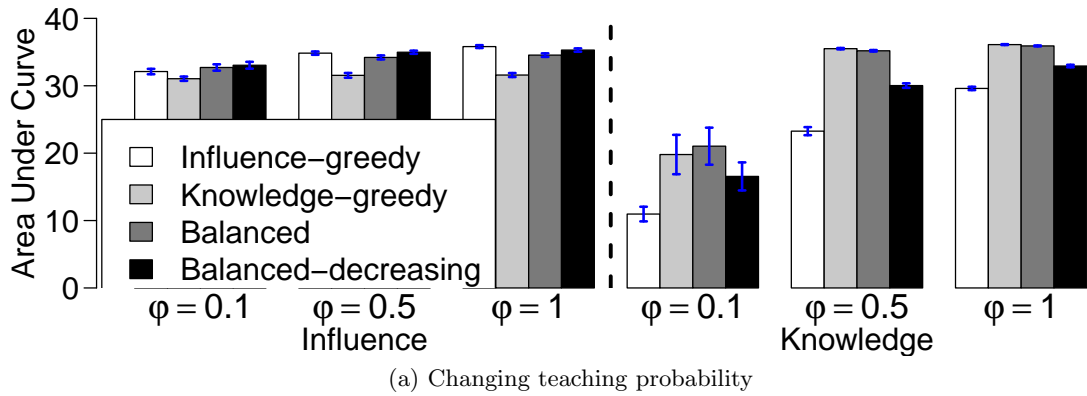
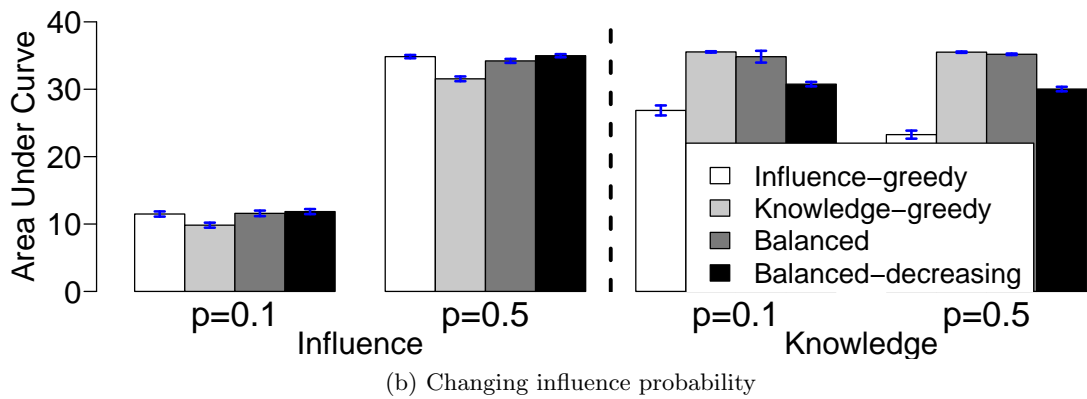


Figure A.13: Results for network A across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming power law distribution.



(a) Changing teaching probability



(b) Changing influence probability

Figure A.14: Results of Influence and Knowledge in network A for different teaching and influence probabilities, assuming power law distribution.

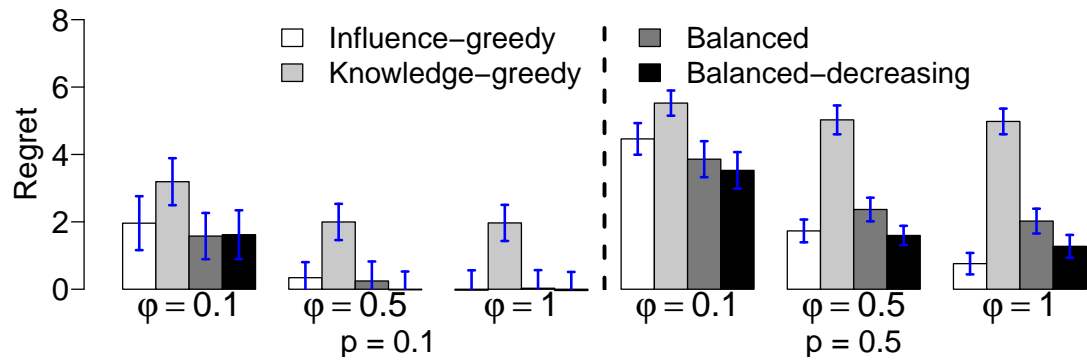


Figure A.15: Regret in network A for different teaching and influence probabilities, assuming power law distribution.

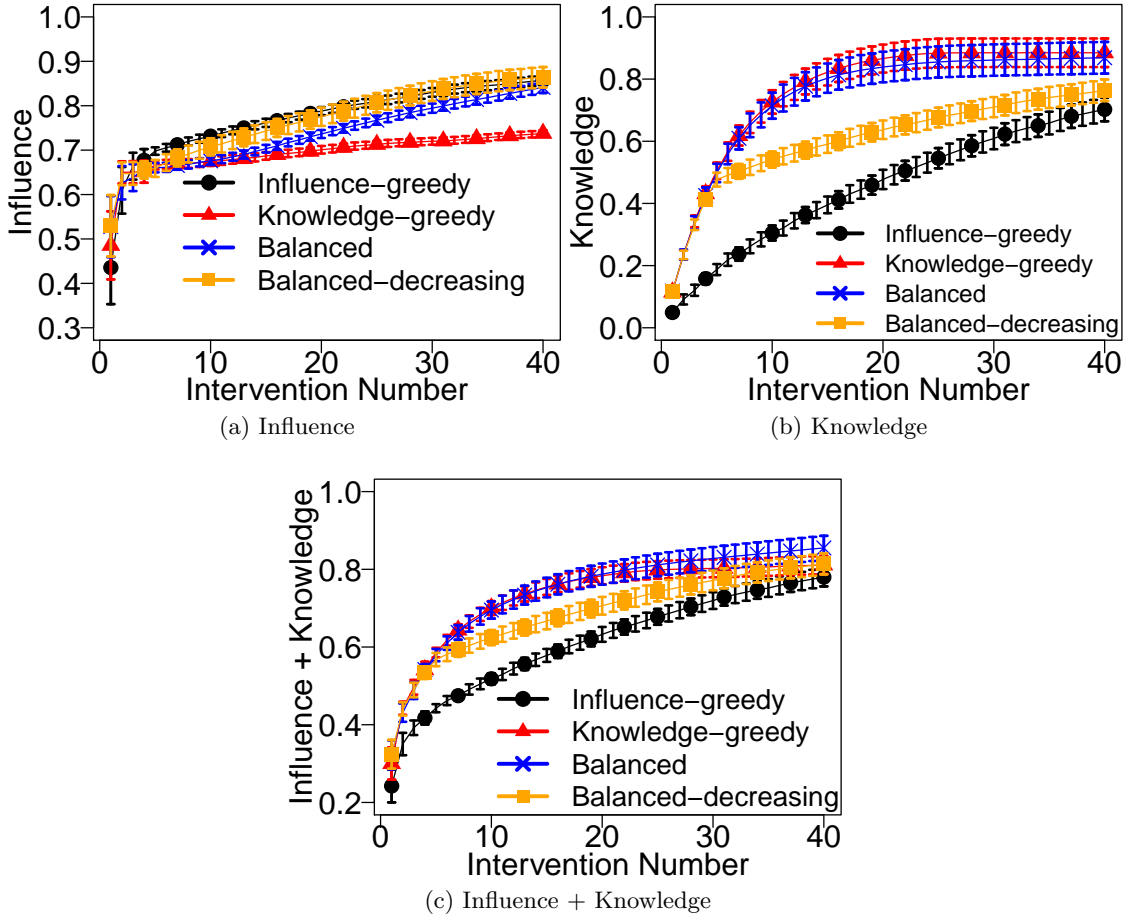


Figure A.16: Results for network B across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming power law distribution.

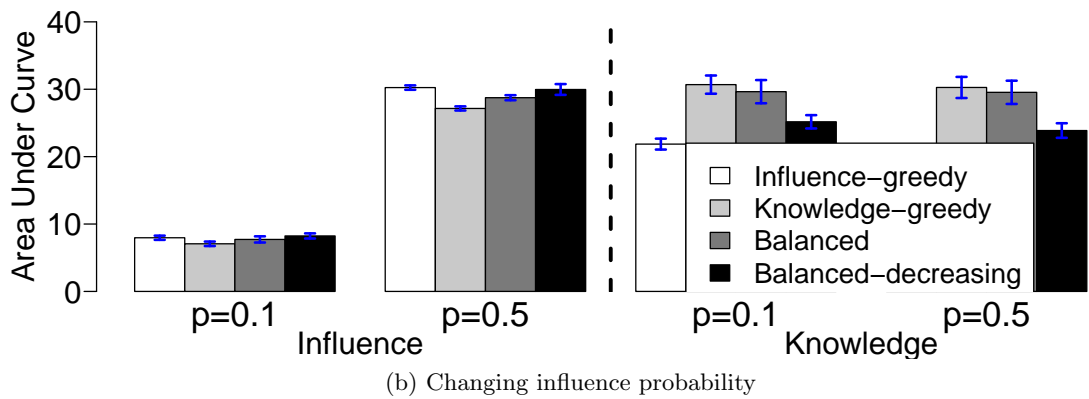
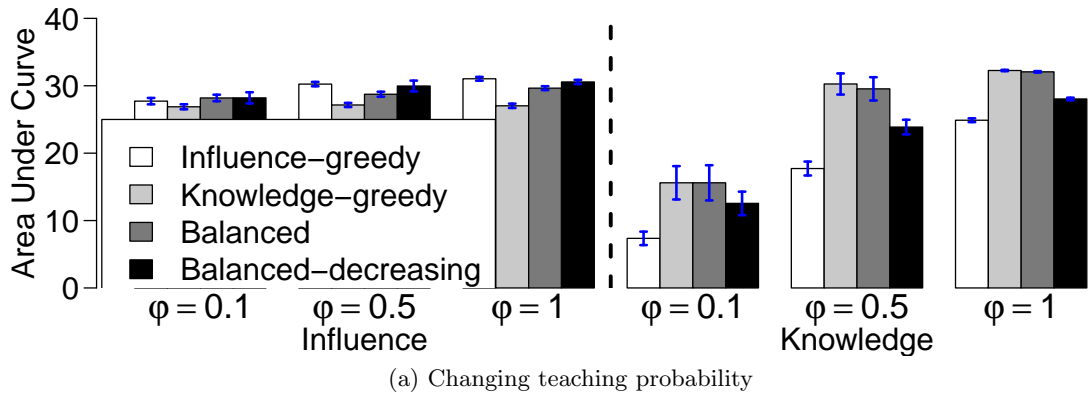


Figure A.17: Results of Influence and Knowledge in network B for different teaching and influence probabilities, assuming power law distribution.

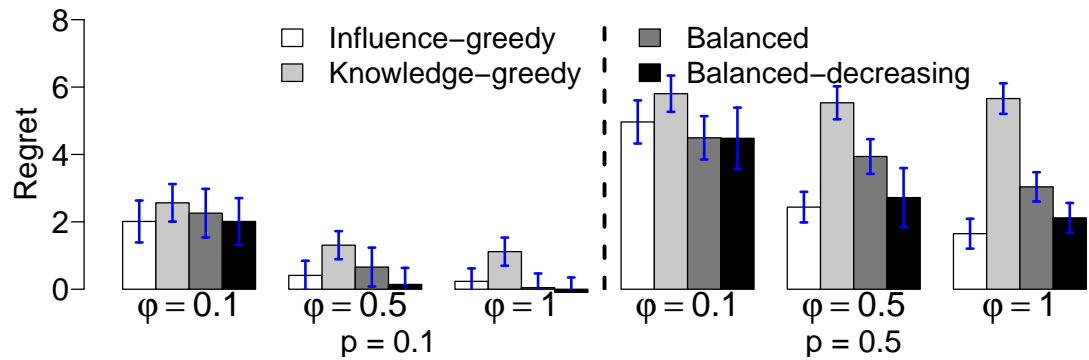


Figure A.18: Regret in network B for different teaching and influence probabilities, assuming power law distribution.

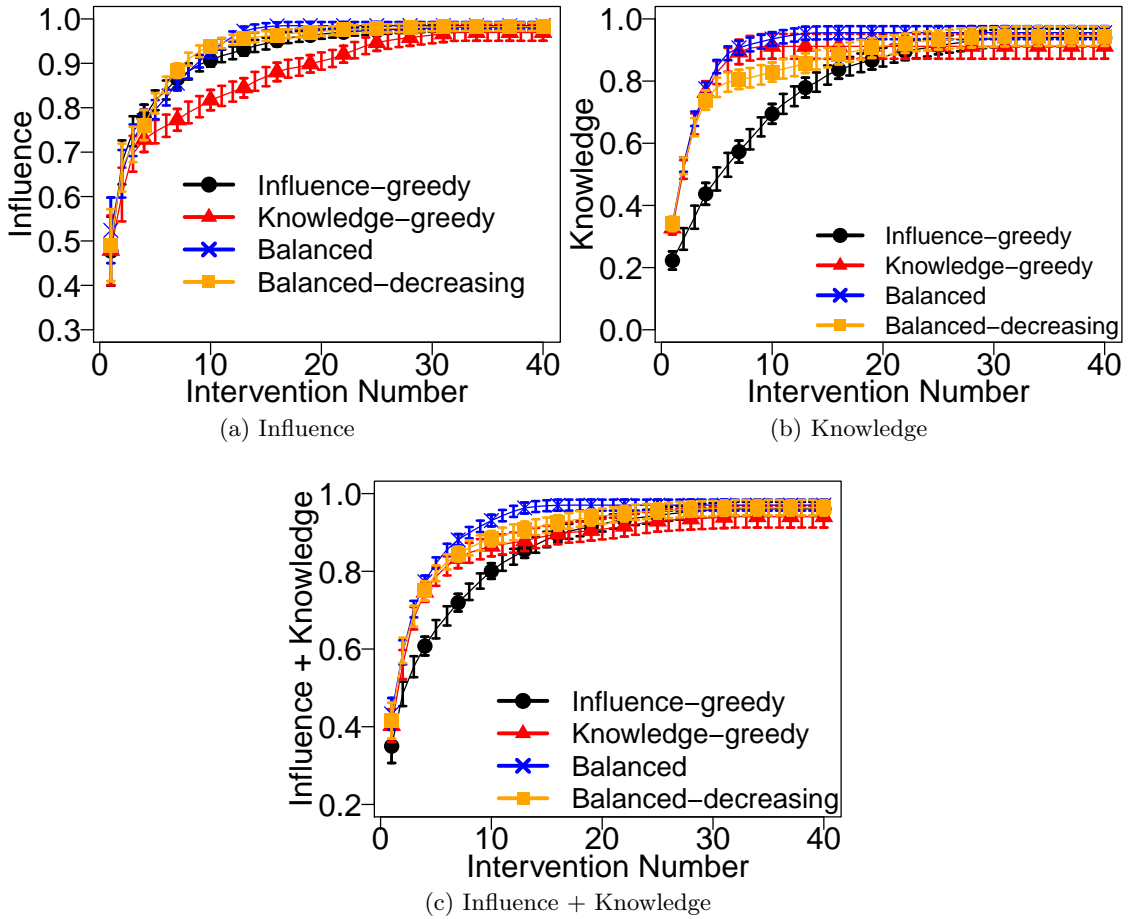
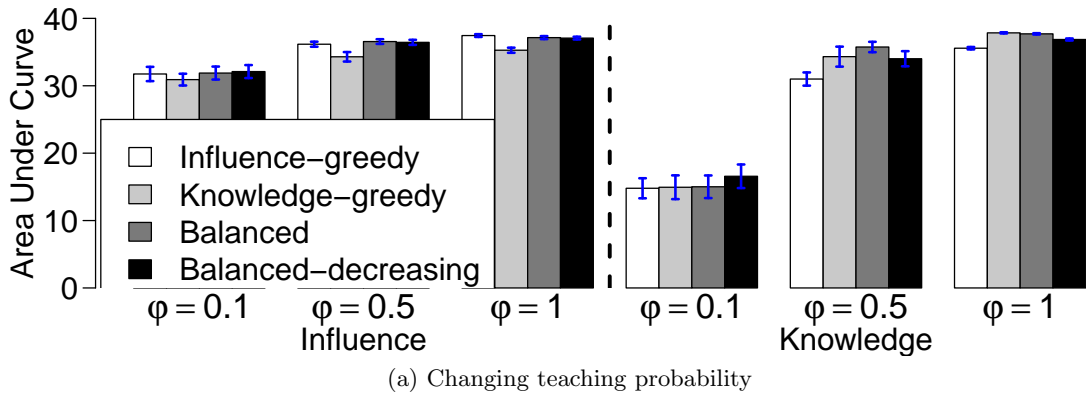
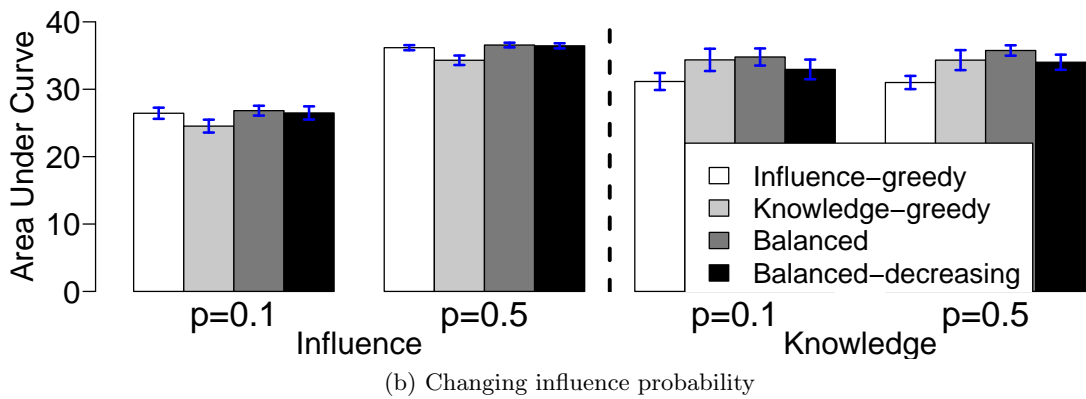


Figure A.19: Results for Facebook network across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming power law distribution.



(a) Changing teaching probability



(b) Changing influence probability

Figure A.20: Results of Influence and Knowledge in Facebook network for different teaching and influence probabilities, assuming power law distribution.

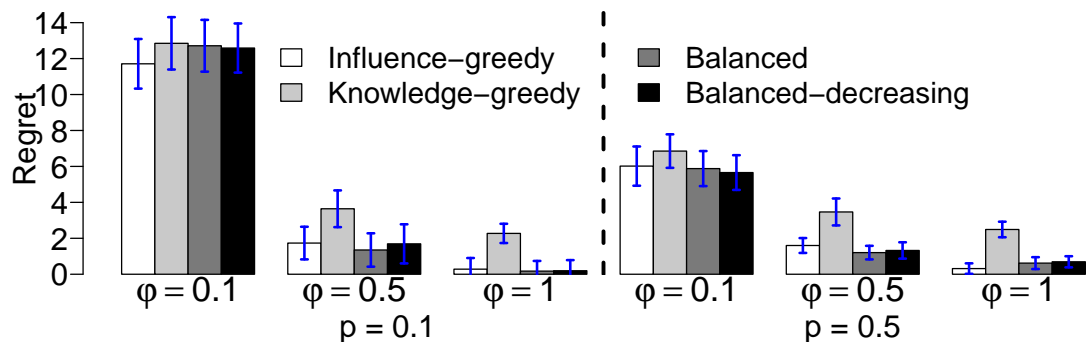


Figure A.21: Regret in Facebook network for different teaching and influence probabilities, assuming power law distribution.

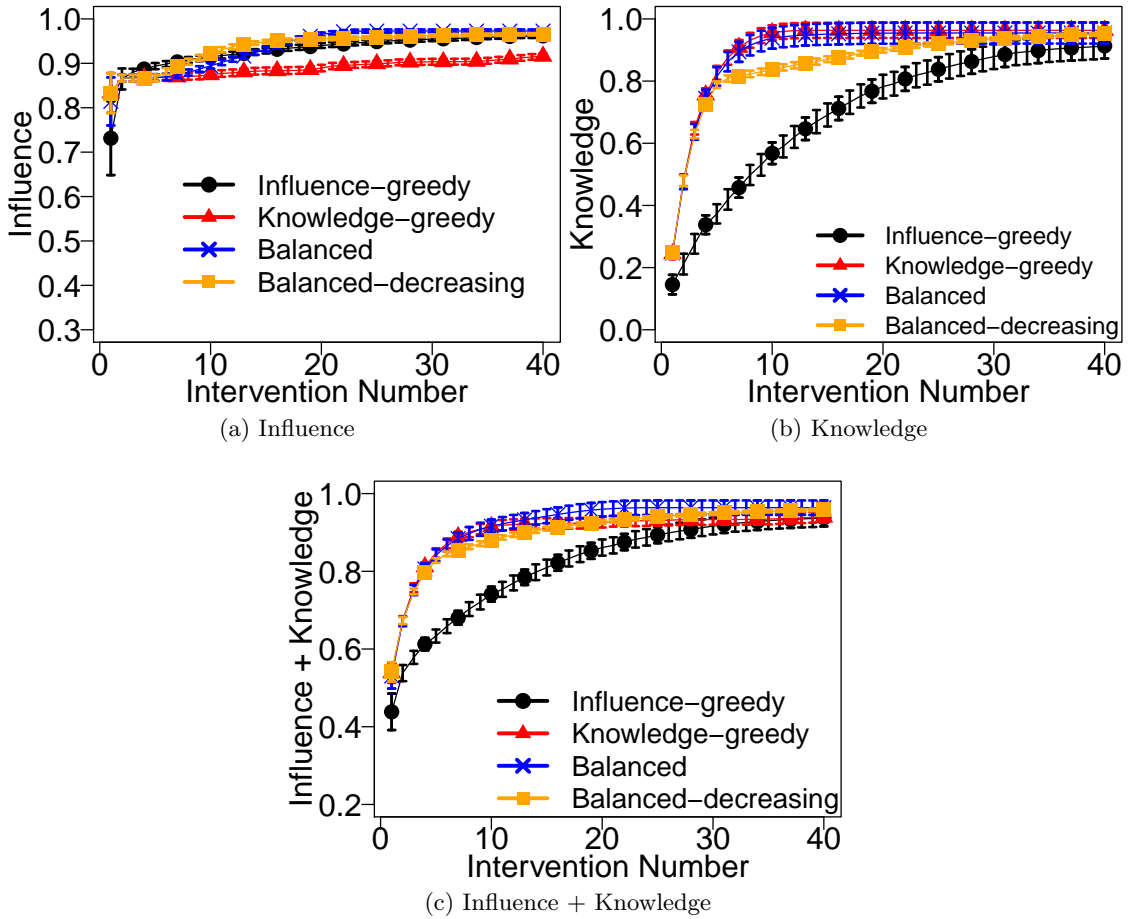


Figure A.22: Results for MySpace network across many interventions, for influence probability $p = 0.5$, teaching probability $\varphi = 0.5$, assuming power law distribution.

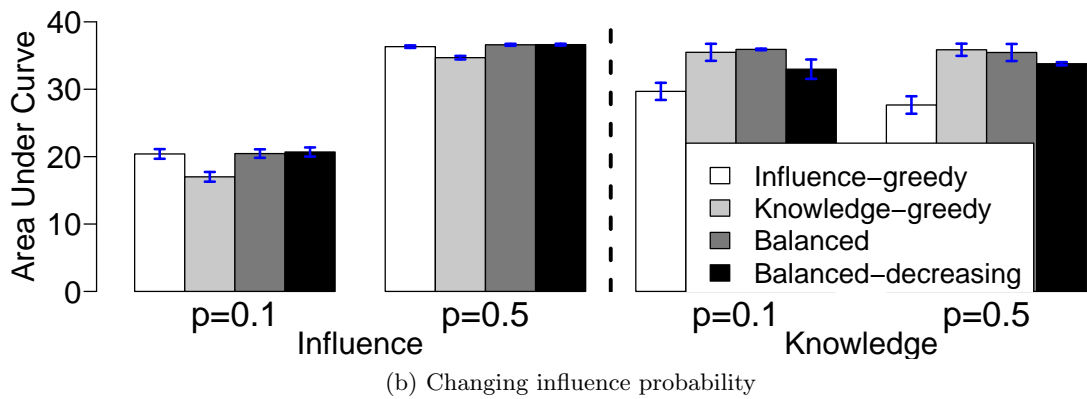
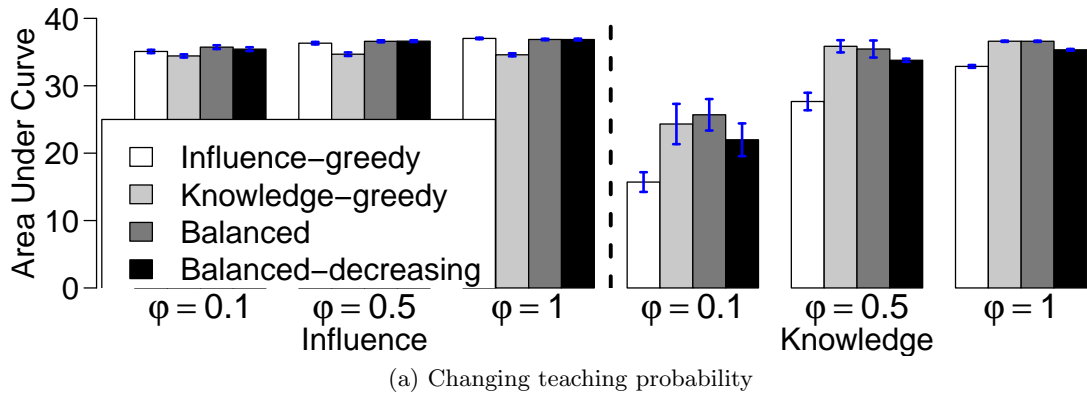


Figure A.23: Results of Influence and Knowledge in MySpace network for different teaching and influence probabilities, assuming power law distribution.

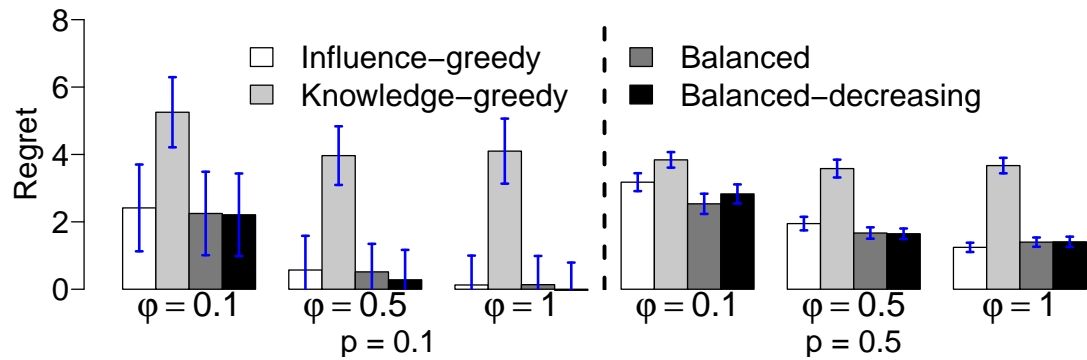


Figure A.24: Regret in MySpace network for different teaching and influence probabilities, assuming power law distribution.

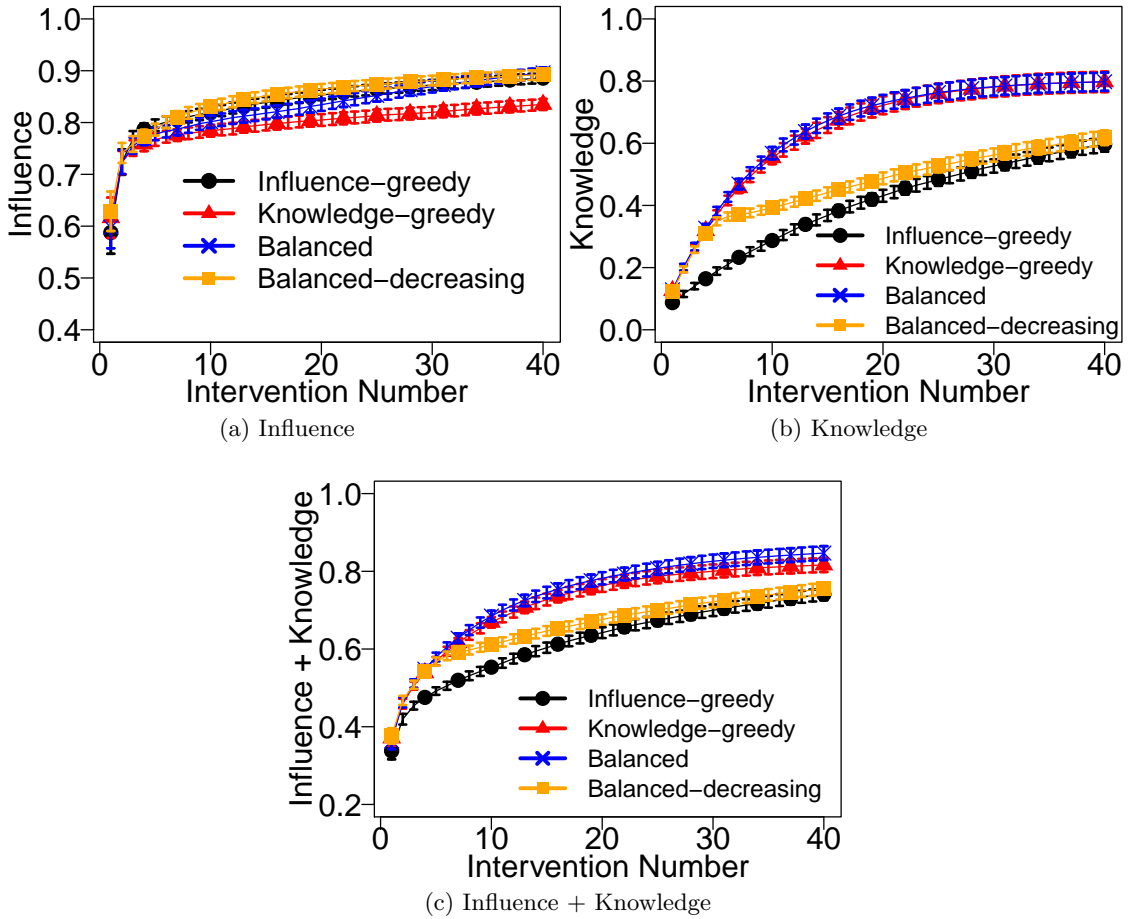
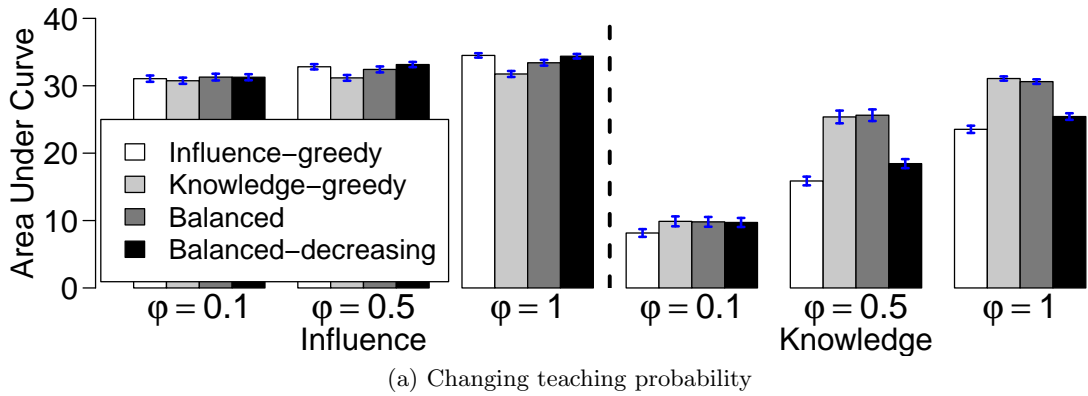
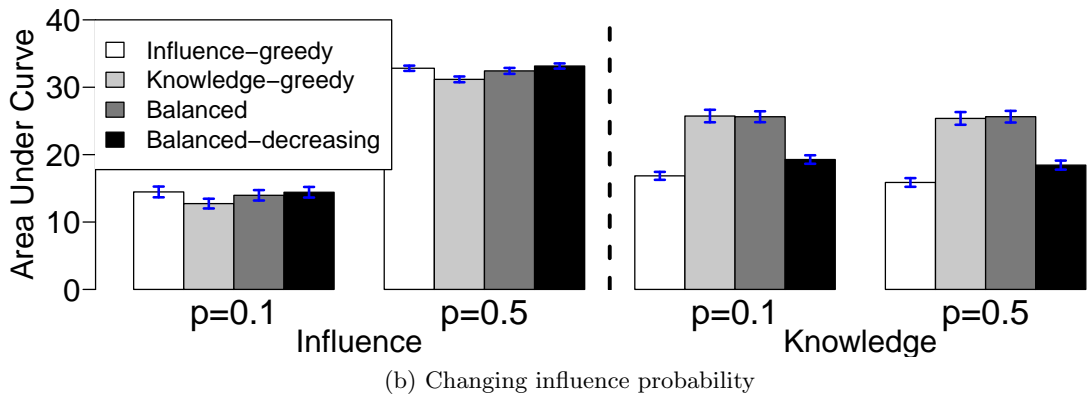


Figure A.25: Results of 4 real world networks across many interventions, for $p = 0.5$ and $\varphi = 0.5$, assuming power law distribution with $a = 1.2$.



(a) Changing teaching probability



(b) Changing influence probability

Figure A.26: Results of Influence and Knowledge for different teaching and influence probabilities, assuming power law distribution with $a = 1.2$.

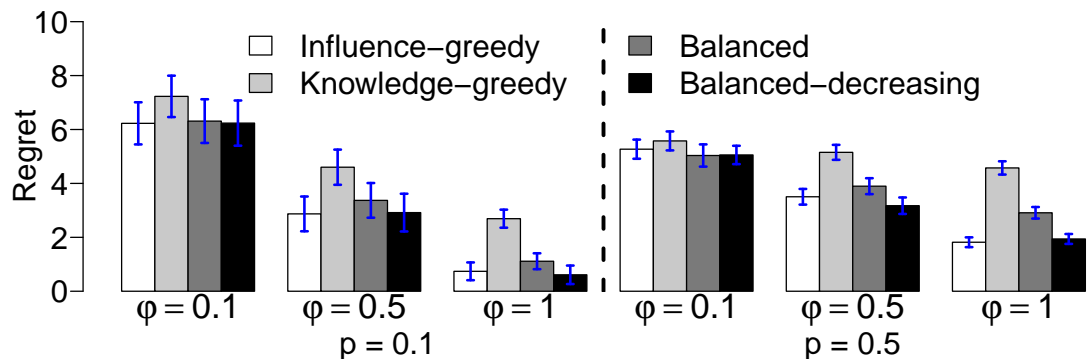


Figure A.27: Regret for different teaching and influence probabilities, assuming power law distribution with $a = 1.2$. Lower results are better.

Appendix B

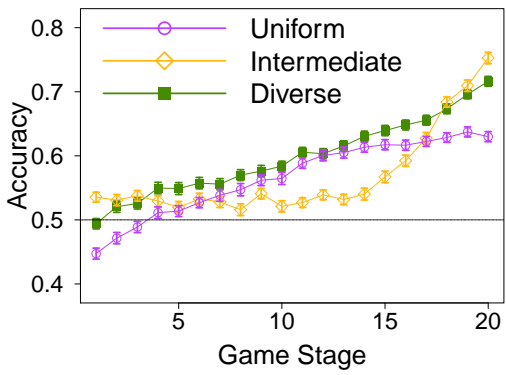
EVERY TEAM DESERVES A SECOND CHANCE: ADDITIONAL RESULTS

In this chapter I present the results of my team assessment methodology (Chapter 8) when comparing with the actual final outcome of the games, instead of using Fuego’s evaluation. I show these results because some readers could be interested in seeing this alternative evaluation methodology.

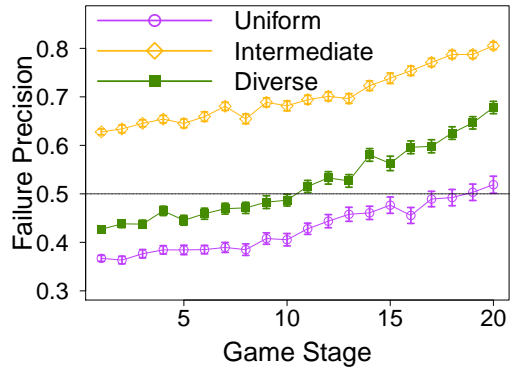
In Figure B.1 I show the accuracy, precision and recall for 9x9 Go games, while in Figure B.2 I show the ROC curves for such games. In Figure B.3 we can see the ROC results for all the 4 different board sizes, while I plot the AUC results with one graph per team in Figure B.4. I also show the AUC results with one graph per board size in Figure B.5. In Figure B.6 I plot the difference between the areas under the AUC graphs for *diverse* and *uniform*, and also the percentage of stages where the prediction for *diverse* is significantly better. Finally, in Figure B.7 I show the accuracy, precision and recall for 21×21 Go (and also 9×9 Go, for comparison). As we can see, the results are similar to the previous ones, and my main conclusions still hold: we still can make better predictions for the *diverse* team than for the *uniform* team, and we can observe a better quality when predicting for 21×21 Go than smaller board sizes. We notice, however, that this time the difference between *diverse* and *uniform* (both in terms of area under the AUC curves and percentage of stages where the AUC for *diverse* is significantly better) is higher on 9×9 Go than on 21×21 Go.

I compare the full and the reduced representation under this alternative baseline as well. In Figure B.8 I show the area under the AUC curves for all teams and board sizes under consideration. Again, the results of both representations are similar¹, but the reduced one is significantly better in almost all cases.

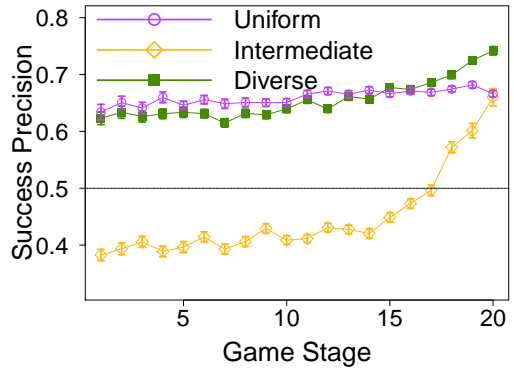
¹My main results still hold, except that the AUC for 21×21 is significantly better than for 9×9 earlier for *uniform* than *diverse* under the reduced representation. For *uniform*, since stage 12, while for *diverse*, since stage 15.



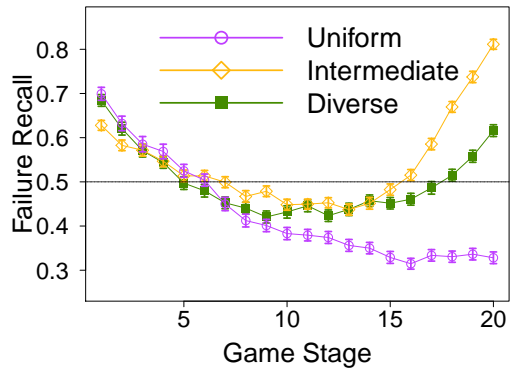
(a) Accuracy



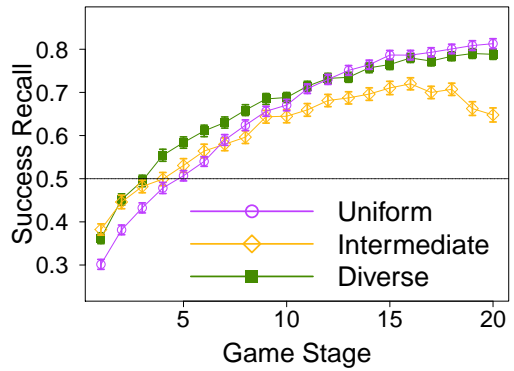
(b) Failure Precision



(c) Success Precision



(d) Failure Recall



(e) Success Recall

Figure B.1: Performance metrics over all turns of 9x9 Go games. (Alternative baseline)

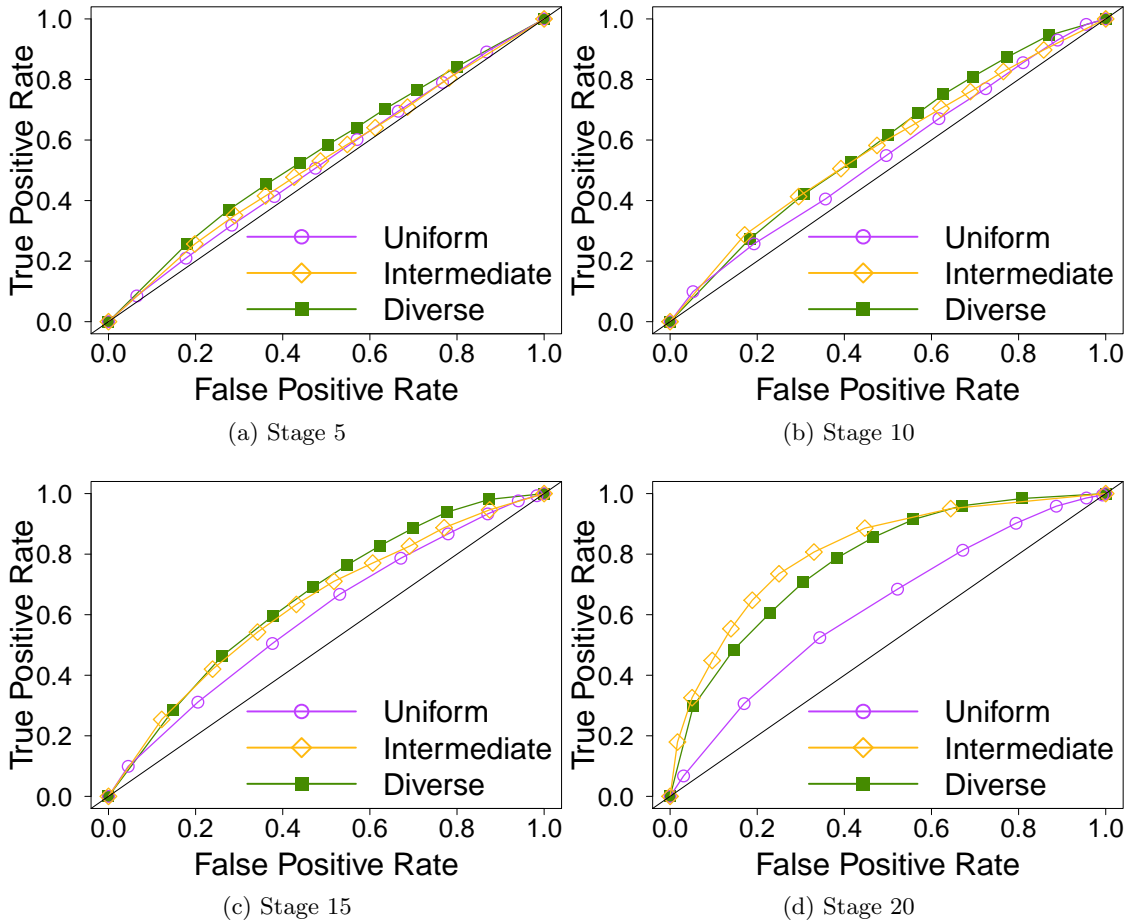


Figure B.2: ROC curves, analyzing different thresholds in 9x9 Go. (Alternative baseline)

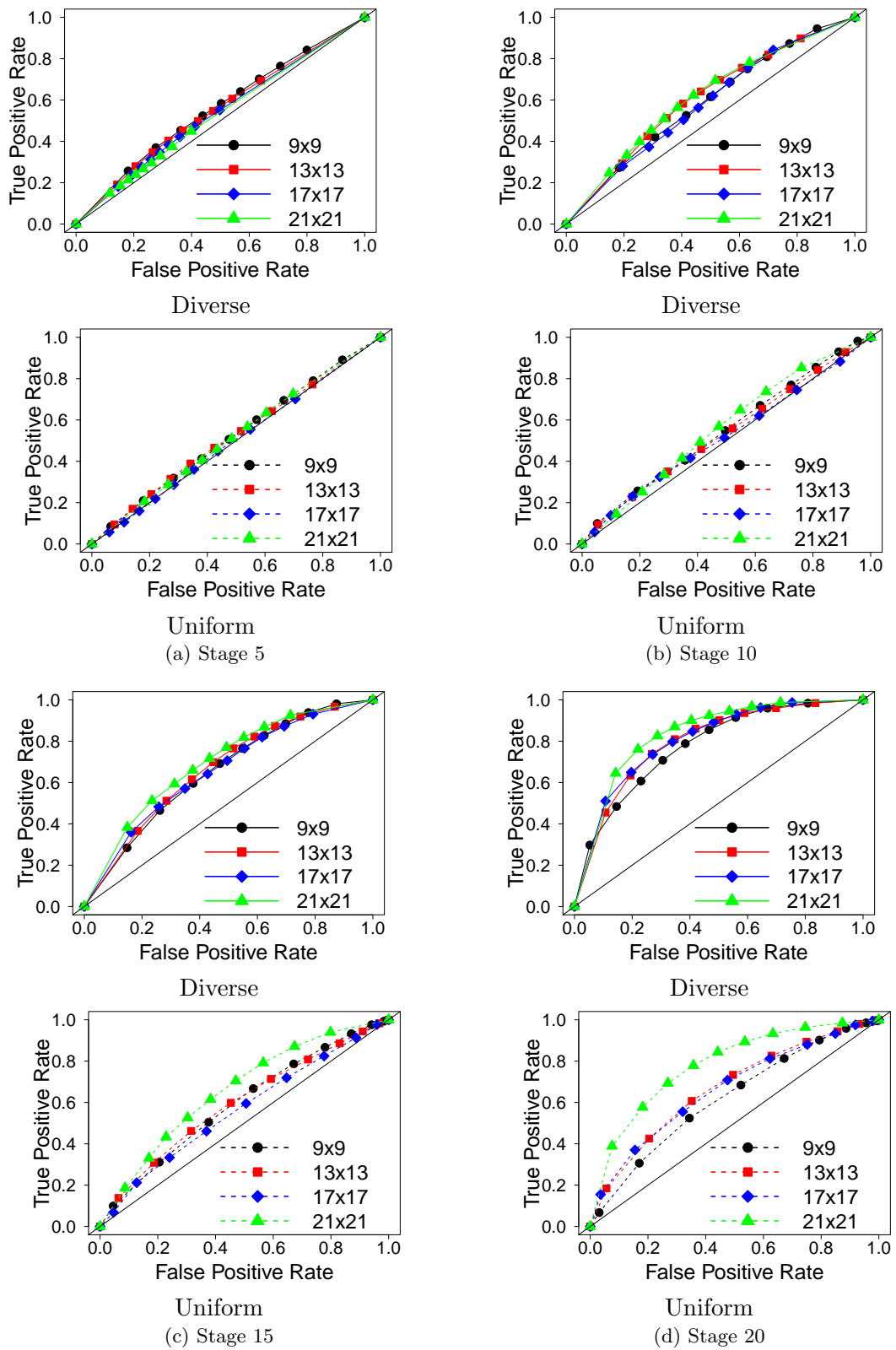


Figure B.3: ROC curves for *diverse* and *uniform*, for different board sizes. (Alternative baseline)

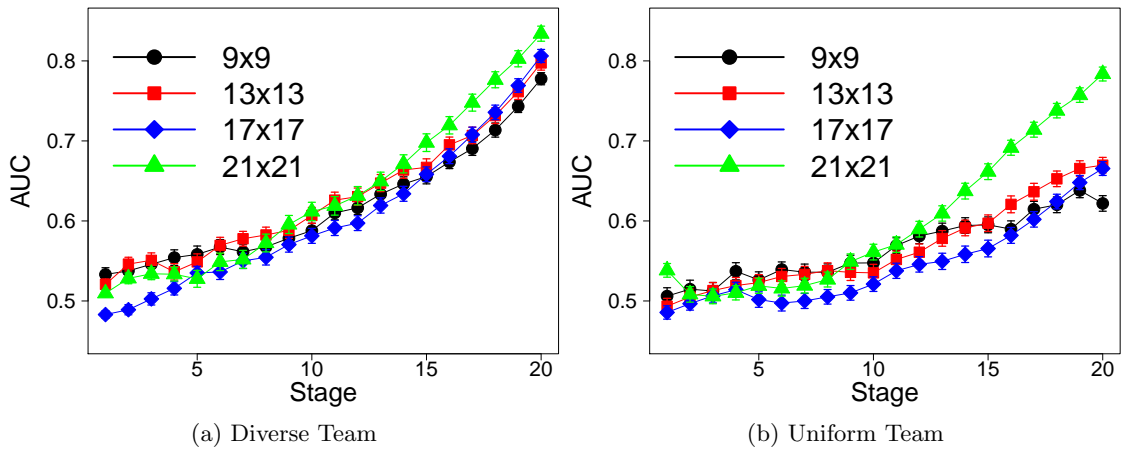


Figure B.4: AUC for different teams and board sizes, organized by teams. (Alternative baseline)

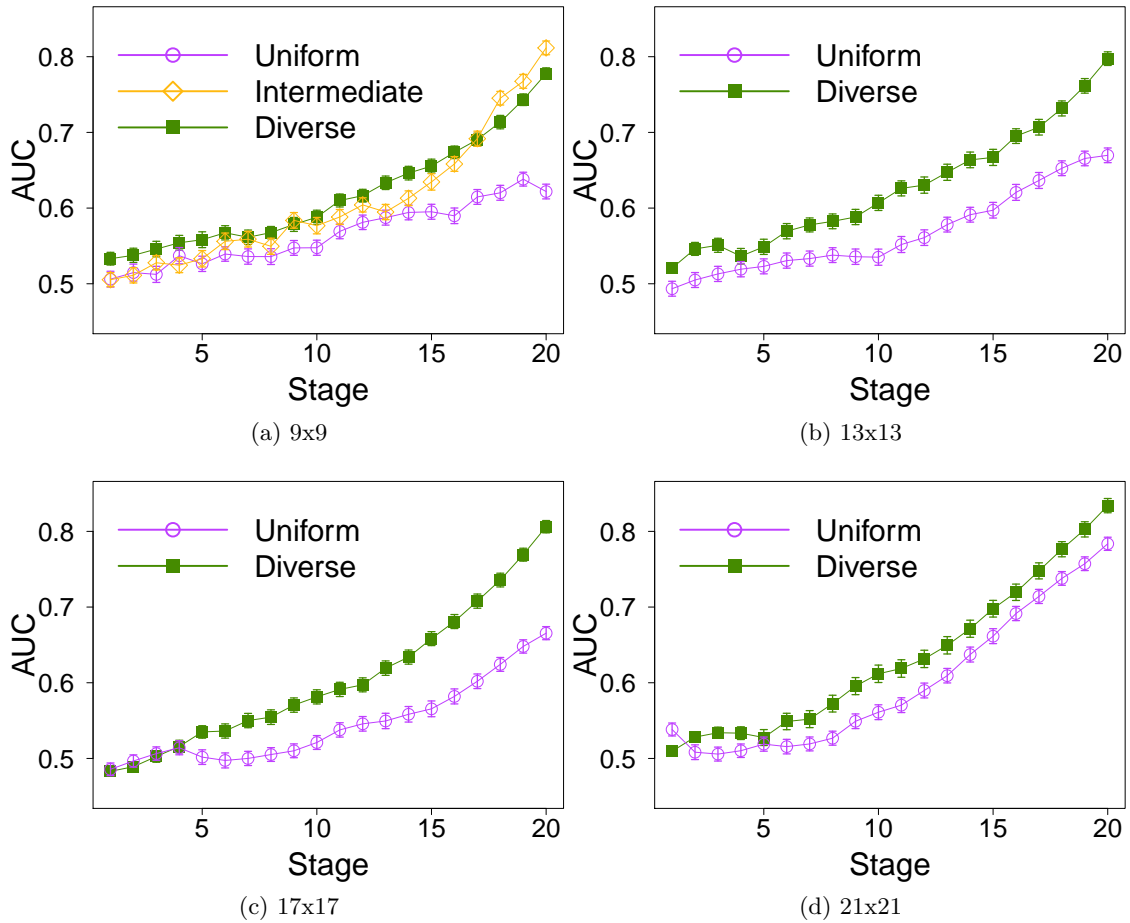


Figure B.5: AUC for different teams and board sizes, organized by board sizes. (Alternative baseline)

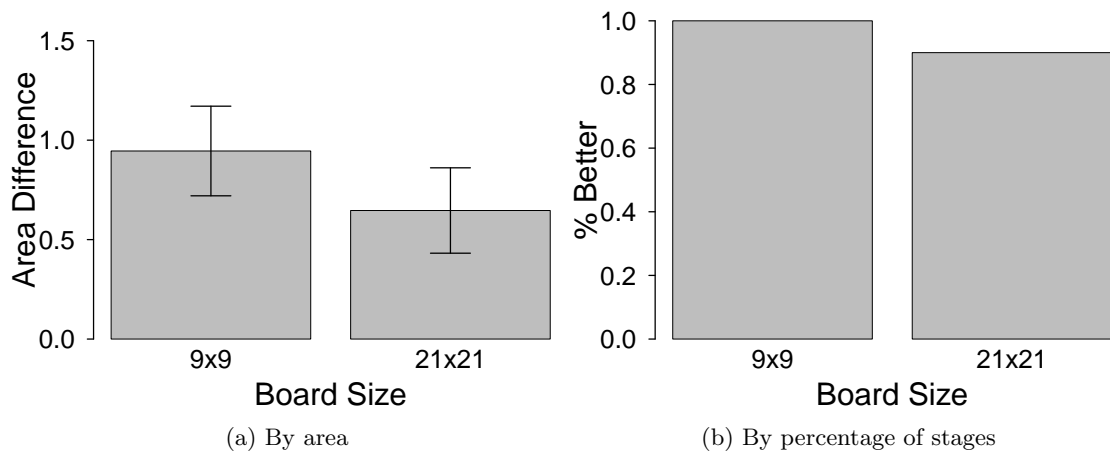
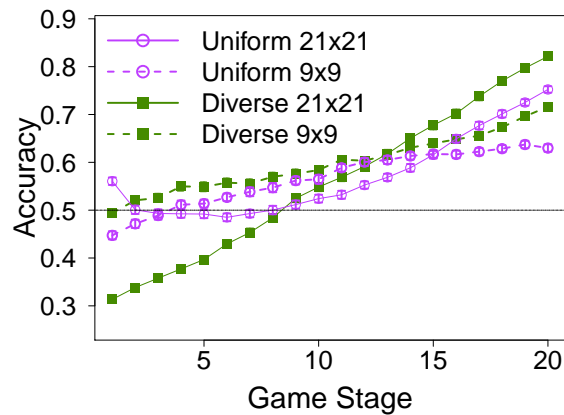
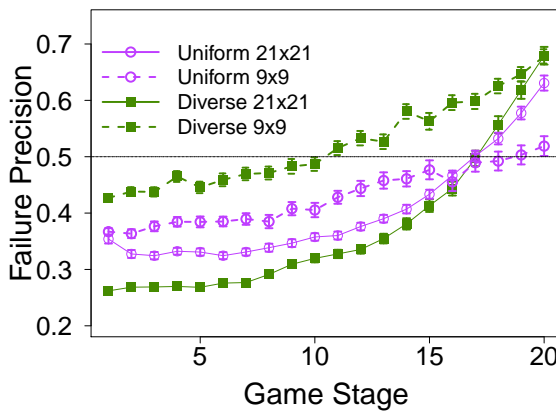


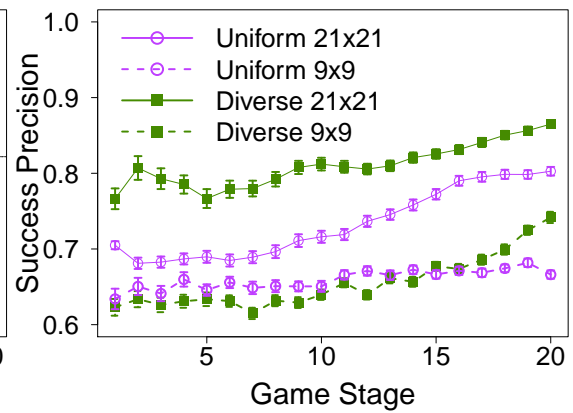
Figure B.6: Differences in prediction quality for the *diverse* and *uniform* teams. (Alternative baseline)



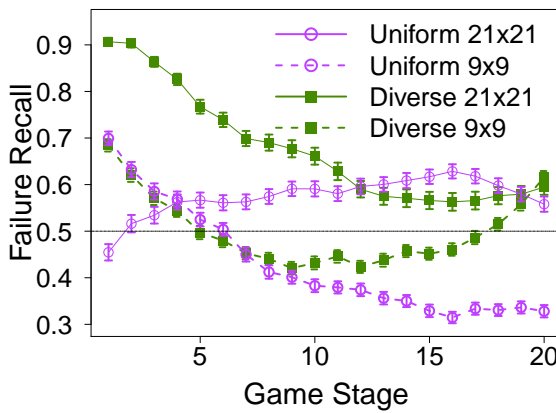
(a) Accuracy



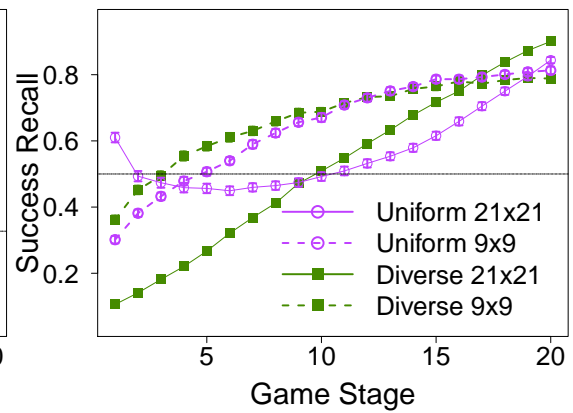
(b) Failure Precision



(c) Success Precision



(d) Failure Recall



(e) Success Recall

Figure B.7: Performance metrics over all turns of 9x9 and 21x21 Go games. (Alternative baseline)

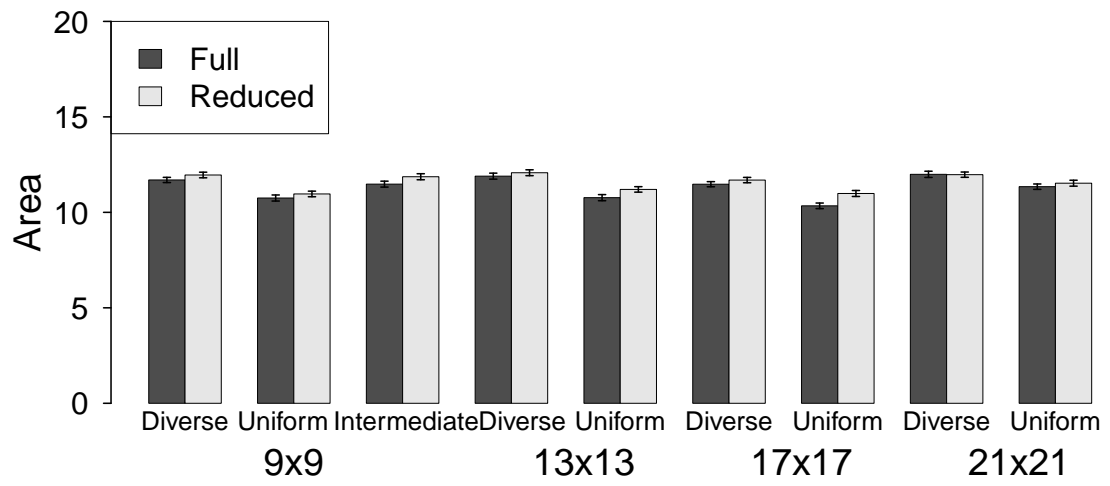


Figure B.8: Comparison of prediction quality with the full and reduced representation. (Alternative baseline)