SAFE GAME OF COMPETITIVE DIFFUSION

by

Celeste Vautour

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science

at

Dalhousie University
Halifax, Nova Scotia
March 2014

Table of Contents

List of Tables . . . . **v**

List of Figures . . . . **vi**

Abstract . . . . **ix**

List of Abbreviations and Symbols Used . . . . **x**

Acknowledgements . . . . **xv**

Chapter 1 Introduction **1**

1.1 Related Work 3

1.2 Notions in Game Theory 7
1.2.1 Games and Strategies 8
1.2.2 Nash Equilibria 12
1.2.3 Safe Games 14

1.3 Notions in Linear Programming 15
1.3.1 Definition of a Linear Programming Problem 17
1.3.2 Solving the Safe Game with Linear Programming 19

1.4 Notions in Graph Theory 21
1.4.1 Common Graphs 21
1.4.2 Properties of Trees 24

Chapter 2 Competitive Diffusion **27**

2.1 Definition of the game 27

2.2 Strategies 28

2.3 Preliminary Results: Nash Equilibrium 30

2.4 Safe Game 31

2.5 Linear Programming 33

Chapter 3 Paths and Spiders **35**

3.1 Paths 35
3.1.1 Game Matrix and Examples 35
3.1.2 Bounds on the Safety Value with the Strategy $C_P(k)$ 37

3.2 Spiders . . . 53
3.2.1 Pure Nash Equilibrium . . . 54
3.2.2 Game Matrix and Examples . . . 55
3.2.3 Bounds on Safety Value with the Strategy $C_{S1}(k)$ . . . 58
3.2.4 Bounds on Safety Value with the Strategy $C_{S2}(k)$ . . . 65

Chapter 4 Complete Trees . . . 72

4.1 Pure Nash Equilibrium . . . 72

4.2 Safety Value of Player 1 on Complete m-ary trees . . . 73

4.3 Special Case: Binary Tree . . . 80

4.4 Special Case: Ternary Tree . . . 80

Chapter 5 Trees in General . . . 82

5.1 Weights of the Vertices in a Tree . . . 82

5.2 Centroidal Trees . . . 85
5.2.1 Types of Branches at the Centroid . . . 85
5.2.2 Centroidal Safe Strategy (CSS) Algorithm . . . 96
5.2.3 Special Case: Centroidal Trees with Only Thick Branches . . . 114
5.2.4 Special Case: Paths, Spiders and Complete Trees Revisited . . 121
5.2.5 Centroidal Safe Strategy Algorithm Performance Assessment on Trees in General . . . 126

5.3 Bicentroidal Trees . . . 151

Chapter 6 A Few Other Graphs . . . 152

6.1 Cycles . . . 152

6.2 Complete Graphs . . . 155

6.3 Complete Bipartite Graphs . . . 157

Chapter 7 Conclusion . . . 163

Appendix A Programming Codes . . . 166

A.1 Generating Centroidal Tree Examples: Method 1 . . . 166

A.2 Generating Centroidal Tree Examples: Method 2 . . . 169

A.3 Centroidal Safe Strategy Algorithm . . . 172

A.4 Centroidal Opposing Strategy Algorithm 176

Bibliography **185**

List of Tables

3.1 Bounds on the safety value of Player 1 on P_n for specific values of n. 53

5.1 Statistics on the difference between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 20,000 centroid tree examples obtained by Method 1 for each pair of parameters $d \in \{2, 3, 5, 10\}$ and $n_1 \in \{100, 1000, 10000\}$. 138

5.2 Statistics on the difference between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a percentage of the weight of the centroid in 1,000 centroid tree examples obtained by Method 2 for $n \in \{100, 1000\}$. 143

5.3 Weights in the centroidal tree of Example 5.35. 145

5.4 Weights in the centroidal tree of Example 5.36. 147

5.5 Mixed strategy for Player 1 on the centroidal tree of Example 5.36. 150

List of Figures

1.1 Expected gains of Player 1 in Example 1.24 when Player 2 chooses HEADS or TAILS. 16

1.2 Examples of common graphs. 22

1.3 Example of weights and centroid of a tree. 25

2.1 Example of a two-player Competitive Diffusion game. 28

2.2 Example of a graph which does not admit a pure Nash equilibrium for the two-player game of Competitive Diffusion. 31

3.1 Labelling of the vertices on a path graph. 35

3.2 Illustration 1 in the proof of Lemma 3.6. 39

3.3 Illustration 2 in the proof of Lemma 3.6. 41

3.4 Illustration 3 in the proof of Lemma 3.6. 43

3.5 Illustration 4 in the proof of Lemma 3.6. 43

3.6 Illustration 1 in the proof of Lemma 3.8. 47

3.7 Illustration 2 in the proof of Lemma 3.8. 48

3.8 Illustration 3 in the proof of Lemma 3.8. 49

3.9 Illustration 4 in the proof of Lemma 3.8. 49

3.10 Labelling of the vertices in a spider with 3 legs. 54

3.11 Illustration of the strategy $C_{S2}(k)$ with $k = 2$ on a spider with 3 legs. 66

4.1 Labelling of the vertices in a complete tree. 74

5.1 Illustration in the proof of Lemma 5.2. 83

5.2 Illustration in the proof of Lemma 5.4. 84

5.3 Example of branches in graphs. 86

5.4 Illustration in the proof of Lemma 5.8. 87

5.5 Possible configurations of the three vertices with lowest weight in a branch at the centroid of a centroidal tree. 88

5.6 Safe strategies of Player 1 on a branch of a centroidal tree with configuration (a) of Figure 5.5. 89

5.7 Safe strategies of Player 1 on a branch of a centroidal tree with configuration (b) of Figure 5.5. 90

5.8 Representation of the CSS Algorithm 109

5.9 Illustration in the proof of Lemma 5.34. 134

5.10 Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 20,000 centroidal tree examples obtained by Method 1 with $d = 2$ branches at the centroid and n_1 vertices in the largest branch. 139

5.11 Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 20,000 centroidal tree examples obtained by Method 1 with $d = 3$ branches at the centroid and n_1 vertices in the largest branch. 140

5.12 Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 20,000 centroidal tree examples obtained by Method 1 with $d = 5$ branches at the centroid and n_1 vertices in the largest branch. 141

5.13 Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 20,000 centroidal tree examples obtained by Method 1 with $d = 10$ branches at the centroid and n_1 vertices in the largest branch. 142

5.14 Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 1,000 centroidal tree examples obtained by Method 2 with the total number of vertices, n. 144

5.15 Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm with the added step suggested from Example 5.35 as a proportion of the weight of the centroid in 1,000 centroidal tree examples obtained by Method 2 with the total number of vertices, n. 148

5.16 Representation of the centroidal tree in Example 5.36. 149

6.1 Illustration of two paths between the vertices in a cycle. 153

6.2 Example of two players choosing starting vertices at equal distance from a cluster in a large graph. 158

6.3 Example of a Complete Bipartite Graph, $K_{m,n}$ with $m = 3$ and $n = 4$. 162

Abstract

Competitive Diffusion is a recently introduced game-theoretic model for the spread of information through social networks. The model is a game on a graph with external players trying to reach the most vertices. In this thesis, we consider the safe game of Competitive Diffusion. This is the game where one player tries to optimize his gain as before, while his opponents' objectives are to minimize the first player's gain. This leads to a safety value for the player, i.e. an optimal minimal expected gain no matter the strategies of the opponents.

We discuss safe strategies and present some bounds on the safety value in the two-player version of the game on various graphs. The results are almost entirely on the safe game on trees, including the special cases of paths, spiders and complete trees but also consist of some preliminary studies of the safe game on three other simple graphs. Our main result consists of a Centroidal Safe Strategy (CSS) Algorithm which suggests a safe strategy for a player on any centroidal tree, a tree which has one vertex as centroid, and gives its associated guaranteed gain.

List of Abbreviations and Symbols Used

Notation	Description
(A, B)	Bimatrix game where the game matrices of the players are A and B.
(I_d, I_e)	Identification of a vertex v_i in a complete m-ary tree in Chapter 4 where I_d is the distance from the root and I_e is the position of v_i in the level I_d.
(I_d, I_s)	Identification of a vertex v_i in a spider in Chapter 3 where I_d is the number of edges from v_i to the body and I_s is the leg it belongs to.
(X^*, Y^*)	Pair of strategies in a saddle point.
(X^A, Y^A), (X^{B^T}, Y^{B^T})	Saddle points of game matrices A and B in the bimatrix game (A, B). X^A is the maxmin strategy of Player 1, Y^A, the minmax strategy of Player 2, X^{B^T}, the maxmin strategy of Player 2 and Y^{B^T}, the minmax strategy of Player 1.
$A = (a_{ij})$	Game matrix of a player in a two player game where a_{ij} is the payoff when the players choose the pure strategies i and j.
A_G	Game matrix of Player 1 in the game Competitive Diffusion on the graph G.
A_j, ${}_iA$	jth column and ith row of the matrix A.
B, B_i	Branches at the centroid in a tree in Chapter 5.
$C(T_n)$	Centroid of the tree T_n.
$C_P(k)$	Mixed strategy on a path described in Definition 3.4.
C_n	Cycle with n vertices.
$C_{S_1}(k)$, $C_{S_2}(k)$	Mixed strategies on spiders with legs of equal length described respectively in Definitions 3.16 and 3.23.
$Cr(B)$	Criterion of the branch B in a centroidal tree in Chapter 5 and described in Definition 5.20.
$E(X, Y)$	Expected gain of Player 1 with the mixed strategy X and Y as the mixed strategy for Player 2.

Notation	**Description**
$E_1(X,Y)$, $E_2(X,Y)$	Expected gains respectively of Player 1 and Player 2 with the mixed strategies X and Y in a bimatrix game.
$G = (V, E)$	Graph with vertex set V and edge set E.
$GGain(G, X)$	Guaranteed gain of Player 1 with the safe strategy X in the Competitive Diffusion on the graph G.
$Gain(G, X, Y)$	Expected gain of Player 1 when she has the mixed strategy X and Player 2 has the mixed strategy Y in the Competitive Diffusion on the graph G.
K_n	Complete graph on n vertices.
$K_{m,n}$	Complete bipartite graph that has m vertices in one bipartition set and n in the other.
$MGain(G, Y)$	Maximal gain of Player 1 against the opposing strategy Y of Player 2 in the Competitive Diffusion on the graph G.
$N(v_i, v_{i+1})$	Number of vertices in branches at v_i other than the one in which lies the centroid and the one in which lies the vertex v_{i+1} for which there exists a $v_{i+1} - v_i - c$ path in a tree in Chapter 5.
P_n	Path with n vertices.
S	Spider graph.
S_k	Strategy set of k pure strategies.
T	Tree graph.
$T(m, h)$	Complete m-ary tree of height h.
T_n	Tree with n vertices.
U, W	Usual variables for the bipartition subsets in a complete bipartite graph.
X, Y, Z	Usual variables for vectors representing mixed strategies. X, usually associated to Player 1 and Y, to Player 2.
$Z(I_d, I_e)$	Equivalent to $Z((I_d, I_e))$, the mixed strategy where a player chooses with probability 1 the vertex (I_d, I_e) and all the other vertices with probability 0 in Chapter 4.

Notation	Description
$Z(I_d, I_s)$	Equivalent to $Z((I_d, I_s))$, the mixed strategy where a player chooses with probability 1 the vertex (I_d, I_s) and all the other vertices with probability 0 in Chapter 3.
$Z(v_k)$	Vector representing a mixed strategy where a player chooses the vertex v_k with probability 1 and the other vertices with probability 0 in the game Competitive Diffusion.
$a_{(I_d,I_s),(J_d,J_s)}$	Payoff to Player 1 in the Competitive Diffusion on a spider when she chooses the vertex (I_d, I_s) and Player 2 chooses the vertex (J_d, J_s).
$\pi_{i,j}\|_n$, π_{ij}	Payoff to Player 1 in the Competitive Diffusion on a path with n vertices when she chooses the vertex v_i and Player 2 chooses the vertex v_j.
α, β, γ, δ, ϵ, η	Usual variables for probabilities assigned on vertices in mixed strategies.
λ_1, λ_2	Mixed strategies on a complete bipartite graph described in Definitions 6.10 and 6.11 respectively.
$\mathcal{O}(1)$	Big O notation.
μ_1, μ_2	Mixed strategies on complete trees described in Definitions 4.3 and 4.4 respectively.
σ, σ_k	Mixed strategies on centroidal trees, σ_k resulting from the CSS Algorithm 5.21.
τ_1, τ_2, τ_3, τ_4	Mixed strategies on centroidal trees described in Definitions 5.9, 5.11, 5.13 and 5.15 respectively.
θ_C	Mixed strategy on a cycle described in Definition 6.1 .
θ_K	Mixed strategy on a complete graph described in Definition 6.6 .
$\xi_{k,l}$	Opposing strategy for Player 2 resulting from the COS Algorithm 5.32.
ζ_1	Mixed strategy on a centroidal tree with only thick branches described in Definition 5.24.
$\{P_1,...,P_p\}$	Set of players in a game.
$\{c\}$, $\{c_1, c_2\}$	Usual variables for the vertices in the centroid of a tree.

Notation	**Description**
$\{s_1, s_2, ..., s_m\}$	Usual set of variables for the legs of a spider in Chapter 3.
${}_BT(n)$	Bicentroidal tree with n vertices.
${}_CT(n)$	Centroidal tree with n vertices.
$argmin_x f(x)$	Argument of the minimum, i.e. the points x for which the function $f(x)$ attains its minimum.
d	Usual variable for the degree of the centroid in a tree in Chapter 5.
e_i	Vector with a 1 in the ith position and zeros elsewhere.
h	Usual variable for the height of a complete tree in Chapter 4.
$l(s_i)$, l	Length of the leg s_i of a spider and usual variable for the length of the legs in a spider with legs of equal length in Chapter 3.
m	Usual variable for the number of legs of a spider in Chapter 3 and the number of branches at the root of a complete tree in Chapter 4.
n	Usual variable for the number of vertices in a graph.
v, u, t, s, t, r	Usual variables for vertices in graphs.
$value(A)$	Value of the game with game matrix A.
$w(v)$, $\overline{w}(v)$	Weight of the vertex v and $n - w(v)$ i.e. the total number of vertices minus the weight of the vertex v.
w_1, w_2, w_3	Lowest three weights in a branch at the centroid in a tree in Chapter 5.
x^*, k^*	Usual notation for an optimal value of a variable.
COS	Centroidal opposing strategy.
CSS	Centroidal safe strategy.
Player 1	Name of first player in the game Competitive Diffusion. (She).

Notation	Description
Player 2	Name of second player in the game Competitive Diffusion. (He).

Acknowledgements

First of all, I want to express my gratitude to my supervisor, Jeannette Janssen, for all her time, support and feedback. You are a kind and understanding person and it was a pleasure working with you.

Secondly, I would like to thank my readers Jason Brown and Richard Nowakowski for their time and comments.

Next, I would like to greatly thank my parents and my brothers for their patience and numerous encouragements and also Luc for all the time he spent listening to me chatter about my research, his suggestions and his belief in me.

Finally, I would like to thank NSERC as well as the Faculty of Graduate Studies and the Mathematics and Statistics Department at Dalhousie University for the funding resources.

Chapter 1

Introduction

Whether we think of Facebook, Twitter or Google+, amongst many other examples of on-line social networks, it does not take too much observation of our surroundings to realize that they are gaining significance in the life of people around us. Within these on-line networks, information, ideas and influence spread from users to users. The spreading of influence is not surprising. The tendency of individuals to resemble one another is a phenomenon that has long been discussed in psychology and sociology. Thus, just as certain toys can become popular amongst kids at school, as political views can become similar within communities or as technologies can become common amongst co-workers, influence occurs in on-line social networks. That is, users of networks can adopt ideas or products based on the choices of their neighbours in the network. Furthermore, influences in on-line social networks can be significant since they do not have regional limitations. On-line social networks provide direct connections to other people no matter the distance of separation or the frequency of encounters in real life.

This can have implications in various contexts. For instance, companies or agencies see this interconnection between people as a useful tool in the launch of new technologies. They can advertise newly developed products by targeting a small number of users of a network and relying on the spread of rumours for their information to reach the other users. This is also known as "viral-marketing". The goal of the companies is to find the initial set of users that will result in the largest span. However, since companies have rivals, the diffusion of products or technologies in social networks is not as simple as one idea spreading from user to user. It rather consists of many conflicting ideas spreading simultaneously. These ideas are competing against each other to take over the largest part of the network. An interesting way to study this type of diffusion is to consider it in a game-theoretic setting. The Competitive Diffusion model introduced recently in [1] has such a setting. In this model, the companies are considered to be players outside a network that play a game in which winning consists of reaching the most users. The model reduces to a game on a graph where the players, with distinct colours, choose a starting vertex. Then, the diffusion of colours occurs step-wise in a first-come, first-served basis. That is, vertices, once coloured, will colour each of their adjacent vertices in the next step and vertices adopt the first colour that reaches them. According to the authors, their game-theoretic model is the first having players outside the network. Other game-theoretic models have considered players as the users of the network with the choice of adopting different products, but this is a completely different game. The decentralized game-theoretic setting of the model lends a new point of focus to the study of diffusion in social networks.

This thesis will focus on the concept of safe game for the Competitive Diffusion on particular graphs. When a player chooses a strategy in the safe game, he asks himself what would be the worst payoff he could receive. In other words, he always assumes that his opponents are out to get him. Thus, the safe game has one player with a goal of maximizing his personal gain while the other players' goals are to limit the gain to the first player instead of maximizing their own. This setting provides safe strategies which have a minimal expected gain no matter the strategies of the opponents. This is interesting in many cases. For example, consider, once again, the diffusion of products from rival companies on a social network. It is likely that a company does not know the marketing strategies of the other companies. In presence of uncertainty, many companies can see a safe strategy as a good choice because it provides knowledge on the worst that could happen. Knowledge of the worst case reduces the risk associated to a decision since the outcome can only be better than assumed. In particular, safe strategies are to be considered by companies which are not risk takers, small companies which cannot afford to take the risk of not getting information about their product to at least a few buyers and companies which feel that their opponents are out to get them. All in all, there is no doubt that the presence of uncertainty about other players is likely to be the case in real life situations and so the safe strategies suggested in this thesis are great available options.

The goal of this chapter is to define the concept of safe game along with some other notions in game theory, linear programming and graph theory essential for the next chapters. A reader already familiar with some of these areas of mathematics could skip the corresponding sections and proceed to Chapter 2 where the Competitive Diffusion will be defined more precisely and some examples of the notions will be given in the point of view of the game considered.

1.1 Related Work

Social networks are a research topic in various fields including mathematics and computer science but also sociology and economics. One can find numerous articles in each of these fields as well as books which combine the views of the different fields (see [12] and [9]). In sociology and economics, we study the characteristic interactions and behaviors of individuals and groups of a social network. From mathematics and computer science comes models that attempt to represent processes in the social networks taking into account these interactions and behaviors. One process being modelled is the diffusion of information or technologies throughout the social network. This process is usually referred to as a "mouth-to-mouth" process, meaning that users learn about the information through their connected peers.

In models for the spread of information, social networks are represented by graphs with vertices corresponding to the users of the network and edges corresponding to links between the users in the network. The vertices, which are initially inactive, become activated once they adopt an idea. There are two general rules for this process. One is that the tendency of a vertex to adopt an idea should increase as more of his neighbours become activated and the other one forces a vertex that becomes activated to stay that way throughout the rest of the process. There are mainly two types of diffusion models i.e. threshold models and cascade models.

In threshold models, vertices become activated once they surpass a certain threshold. The core of these models is the Linear Threshold Model. In this model, a vertex v is influenced by each of its neighbours, w, by a weight $b_{v,w}$, where the weights $b_{v,w}$ are such that

$$\sum_{w \text{ a neighbour of } v} b_{u,w} \leq 1. \tag{1.1}$$

That is, the extend to which a vertex is influenced by a neighbour is represented by the weight of the edge that connects the vertex to its neighbour. Weights are rescaled so that the total weight of all edges incident with a vertex never exceeds 1. Thus, the weights can be interpreted as probabilities. Moreover, each vertex v has a threshold θ_v uniformly and randomly chosen from the interval $[0, 1]$. The different possible values for the weights and the thresholds of the vertices translate the fact that some people are more easily influenced than others. Given an initial set of active vertices, the diffusion progresses in steps; any vertex v for which the total weight of its active neighbours is at least θ_v becomes activated. In other words, after each step,

$$\sum_{w \text{ an active neighbour of } v} b_{u,w} \tag{1.2}$$

is computed for every inactive vertex v. The vertices for which this value reaches its designated threshold become activated in the current step.

In cascade models, as a vertex becomes activated, it activates each of its neighbours with a given probability. The core of these models is the Independent Cascade Model. In this model, we also start with an initial set of activated vertices. Then, the diffusion progresses in steps where a vertex v that becomes activated, in turns, tries to activate each of its neighbours, w and succeeds with a probability $p_{v,w}$. The probabilities $p_{v,w}$ are parameters of the model. An important aspect of the process is that, once a vertex succeeds or fails in activating its neighbours, it does not attempt again for the rest of the process. In particular, the Competitive Diffusion from [1] on which this thesis is focused is a cascade model where the probability of activating a neighbour is 1.

Moreover, the study of diffusion models is divided into several settings. One of them is an optimization problem. In this setting, the diffusion process can be thought of as a fire that spreads through the network. The course of the fire can follow either a threshold or cascade model of diffusion. The problem consists of determining where to set the fire so that a maximum number of vertices get burned. In other words, we are looking for the set of initial influenced users which will bring a greater overall influence throughout the network. One can find results on this problem in [14], [4] and [25]. In these articles, we also find variations of this problem. For instance, we can have more than one product being diffused on the same network. In this case, the optimization is in the point of view of the last person choosing where to light his fire, once he knows where the others have been lit. This setting resembles the

Competitive Diffusion of [1] by having a point of focus outside the social network. However, it is not regarded as a game-theoretic setting where players simultaneously choose strategies and live the outcome.

Another approach to study diffusion models is a game-theoretic one. First, we have what are called network coordination games. In these games, the players are the users of the network. They receive payoffs in relation to the similitude with their neighbours in the adoption of products or ideas. For example, consider the network coordination game in which each vertex has a choice between two possible products, A and B. If the vertices v and w are adjacent in the network, we can have payoffs as follows

- If both the vertex v and the vertex w adopt the product A, they each get a payoff of $\alpha > 0$;
- If both the vertex v and the vertex w adopt the product B, they each receive a payoff of $\beta > 0$;
- If the vertex v and the vertex w adopt distinct products, they receive a payoff of 0.

Choosing technologies is an example of a real life situation where this could occur. It is advantageous for a person to have technological devices which are compatible with the ones of his friends. Moreover, the larger number of friends with whom he's compatible, the larger the reward. Thus, the resulting payoff of a vertex, v will be the sum of payoff over all its edges. Knowing the values of α and β, we can calculate the fraction of neighbours of v needed to cause him to change products. That is, the fraction of neighbours which would result in a higher resulting payoff if he switches. Thus, the diffusion in this model has vertices switching products once it is to their advantage. Many questions on this model have been looked at such as finding the set of initial vertices which should be chosen by a new product B trying to influence its way through a network which is initially influenced by the product A, describing the Nash equilibria of coexistence between products A and B and studying the characteristics of clusters which break the spread of influence. Much literature concerning these games can be found in [11], [6] and [27]. In these articles, one can also observe variations in the number of products to be chosen from, the added possibility of the users to adopt two products with additional cost and distinct values of payoffs, α and β, for each user. The difference between these models and the Competitive Diffusion of [1] is clearly the position of the players in the game.

Alternatively, there are the game-theoretic settings in which the players are considered to be outside the social networks. They choose initial users to influence and their goal is to reach the most users. The Competitive Diffusion of [1] (see also erratum [31]) is known to be the first model of this sort. In their article, the authors discuss the relationship between the diameter of the graph and the existence of pure Nash equilibria. In [30], the existence of a Nash equilibrium for this game on trees is shown. Moreover, [29] considers the model of Competitive Diffusion on a recently proposed model for on-line social networks and discusses the existence of Nash equilibria.

Since the appearance of the Competitive Diffusion, a couple of generalizations have been looked at. [10] and [34] both generalize the Competitive Diffusion model by having the players in the game-theoretic setting outside the network, but with different diffusion processes. In [10], the agents choose an allocation of budgeted seeds over the vertices and the diffusion process is stochastic. The influence of a user is divided into two parts. First, a switching function, in terms of the number of his neighbours which have adopted either one of the products, gives the probability of a user of adopting a product. Then, there is a selection function, which is conditional upon switching and which gives the probability of the user adopting a product given the proportion of his neighbours with it. The diffusion is divided in discrete time frames where all inactivated vertices either update simultanously or are randomly chosen to update. The authors consider mixed strategies for the players. That is, the players assign probabilities to each possible distribution of seeds on the vertices of the graph, given the total number of seeds in their budget. The authors explore the existence of mixed Nash equilibria with different switching and selection functions. In [34], the agents choose an initial set of k vertices and the diffusion is a threshold model. The model includes two tie-breaking criterions, one for when a vertex is initially chosen by two players and one for when a vertex is eligible to adopt more than one product in the diffusion process. The tie-breaking criterions used are a common reputation ordering of the products, i.e. a product with a higher reputation being adopted over one with a lower reputation. Here, the authors study the existence of pure Nash equilibria for different variations of the threshold model.

The focus in all these articles is finding and describing Nash equilibria. A Nash equilibrium is a state in which all the players know the strategies of the other players and no player can increase his personal gain by changing his strategy while his opponents keep theirs unchanged. Thus, we assume that each player has the full information on what his opponents are doing. In real life situations, this is improbable. In contrast, the safe game has players with no knowledge of their opponent's strategies. In fact, when adopting a strategy, a player assumes the worst possible outcome. This gives safe strategies with guaranteed gains no matter the strategies of the opponents. Hence, the game is more representative of real life situations. The study of the Competitive Diffusion in a safe game setting shows relevance and constitutes the point of focus of this thesis.

Parallel to the game-theoretic setting, we also have Voronoi games (see for example [7], [21]). In these games, the players or facilities choose locations in the space and points or customers that are closest to the facilities are the players' payoffs. The version of this game which seems the most analogous to the Competitive Diffusion is the Voronoi game on graphs (see [32] and [2]). Here, the players take turns choosing vacant vertices on a graph to occupy. This is repeated for n rounds. The distance between two vertices is the number of edges in the shortest path between them. As the game progresses, the vertices are being dominated by the nearest occupied vertex. In the end, the payoffs of the players is the total number of vertices they dominate. Although being similar, the Competitive Diffusion brings a more complex diffusion process than assigning the vertices to the players only based on distances in the graph.

The game-theoretic setting of the Competitive Diffusion of [1] where the players are considered to be outside the network is still a new concept that lends a new point of focus for the study of diffusion models through social networks.

1.2 Notions in Game Theory

Game theory is a branch of mathematics that analyzes strategies for individuals in competitive situations where their outcome depends on the actions of their opponents. One can find many books introducing the basic notions of game theory (e.g., [3]), but all the necessary concepts for this thesis will be presented in the following sections.

1.2.1 Games and Strategies

We should start by defining a game in the sense of classical game theory.

Definition 1.1. A **game** involves a number of players, a set of strategies for each player and a payoff that quantitatively describes the outcome of each play of the game in terms of the amount that each player wins or loses.

Definition 1.2. A **pure strategy** of a player is a strategy that completely defines how a player will play in a game. The set of pure strategies available to a player is called a player's **strategy set**.

Let us consider as an example, a simple two-player game: **Matching Pennies**.

Example 1.3. In the game Matching Pennies, both players have a penny that they must secretly turn heads or tails. The players then reveal their choice at the same time. If the pennies match (both being heads or both being tails), Player 1 wins a point and Player 2 loses a point. If the pennies do not match (one penny is head and the other is tail) Player 2 wins a point and Player 1 loses a point.

In this game, there are two pure strategies available to both players: turning the penny heads or turning the penny tails. Thus, the strategy set of both players is: $\{\textsc{Heads}, \textsc{Tails}\}$.

Suppose we have a two-player game and the players are called Player 1 (She) and Player 2 (He). Say Player 1 has a choice amongst n possible strategies and Player 2 has a choice amongst m possible strategies. Let a_{ij} and b_{ij} be the payoffs of Player 1 and Player 2 respectively if Player 1 chooses the pure strategy i, $1 \leq i \leq n$ and Player 2 chooses the pure strategy j, $1 \leq j \leq m$. Then $\{a_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}$ and $\{b_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}$ are two collections of numbers that can be arranged in matrices to form the payoff or game matrices of each player.

Definition 1.4. The **game matrix** of a player is a matrix $A = (a_{ij})$, $1 \leq i \leq n$, $1 \leq j \leq m$ of real numbers such that the entry a_{ij} represents his payoff if he chooses to play the strategy associated to row i and his opponent chooses to play the strategy associated to column j.

Example 1.5. In the game Matching Pennies described in Example 1.3, the game matrices of Player 1 and Player 2 are

$$A = \begin{array}{c} \\ \textsc{Heads} \\ \textsc{Tails} \end{array} \begin{array}{c} \begin{array}{cc} \textsc{Heads} & \textsc{Tails} \end{array} \\ \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \end{array}, \qquad B = \begin{array}{c} \\ \textsc{Heads} \\ \textsc{Tails} \end{array} \begin{array}{c} \begin{array}{cc} \textsc{Heads} & \textsc{Tails} \end{array} \\ \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \end{array}.$$

Definition 1.6. A **zero-sum game** is a game with two or more players, in which the gain of one player balances the loss of the other players and vice-versa. That is, in a play of a zero-sum game with k players, if the payoff to a player is g_1 and the payoffs to his opponents are g_2, g_3,...,g_k, for $k \in \mathbb{Z}$, then $\sum_{i=1}^{k} g_i = 0$.

In a two-player zero-sum game, whatever one player wins, the other loses so if a_{ij} is Player 1's payoff when Player 1 plays the strategy associated to row i and Player 2 plays the strategy associated to row j, then $-a_{ij}$ is Player 2's payoff for the same play. This means that in the case of a zero-sum game, the game matrices of Player 1 and Player 2 are related i.e. $B = -A$. Thus, all the information on the payoffs can be determined from the game matrix of Player 1.

Example 1.7. The game Matching Pennies described in Example 1.3 is a zero-sum game. Thus, the game matrices presented in Example 1.5 have the property $A = -B$.

Definition 1.8. A **mixed strategy** for a player is a vector $X = (x_1, x_2, ..., x_n)$ such that $x_i \geq 0$ for all $1 \leq i \leq n$ and $\sum_{i=1}^{n} x_i = 1$. The entry x_i in the vector is the probability that the pure strategy associated to row i of the game matrix will be chosen by the player.

Example 1.9. A mixed strategy for Player 1 in the game Matching Pennies described in Example 1.3 could be $X = \left(\frac{2}{3}, \frac{1}{3}\right)$, i.e. choosing to turn her penny Heads with probability $\frac{2}{3}$ and Tails with probability $\frac{1}{3}$.

Let us denote the set of mixed strategies available to a player from a strategy set of k pure strategies by $S_k = \{(z_1, z_2, ..., z_k) \mid z_i \geq 0, 1 \leq i \leq k, \sum_{i=1}^{k} z_i = 1\}$.

Remark 1.10. Pure strategies are part of the set of mixed strategies, S_k. Precisely, a pure strategy is of the form $e_i = (0, ..., 0, 1, 0, ..., 0)$, $1 \leq i \leq k$, where the probability of 1 in the ith position corresponds to the row i or the column i always being played by the player.

When considering mixed strategies, the payoffs of the players become expected payoffs or gains.

Definition 1.11. Consider the two-player game with A as the game matrix for Player 1. Suppose Player 1 chooses the mixed strategy $X \in S_n$ and Player 2 chooses independently the mixed strategy $Y \in S_m$. The **expected payoff** of Player 1 is

$$\begin{aligned} E(X,Y) &= \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} \cdot P(\text{Player 1 chooses } i \text{ and Player 2 chooses } j) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} x_i a_{ij} y_j = XAY^T. \end{aligned} \tag{1.3}$$

where Y^T is the transpose of the vector Y.

The expected gain of Player 1, $E(X,Y)$ is what Player 1 can expect to receive when the game is played many times with the strategies X and Y.

Example 1.12. In the game Matching Pennies described in Example 1.3, suppose Player 1 chooses the mixed strategy $X = \left(\frac{2}{3}, \frac{1}{3}\right)$ while Player 2 chooses the mixed strategy $Y = \left(\frac{3}{4}, \frac{1}{4}\right)$. Player 1's expected gain can be calculated with the formula from the previous definition:

$$E(X,Y) = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} \frac{3}{4} \\ \frac{1}{4} \end{pmatrix} = \frac{1}{6}.$$

Now, let us consider the expression of the expected gain in the previous definition if one or both players choose a mixed strategy equivalent to a pure strategy.

Remark 1.13. Let A be the game matrix of Player 1 in a two-player game and let ${}_iA$ denote the ith row of A and A_j denote the jth column of A. Consider the following two cases, in which the expression for the expected gain can be simplified:

(i) One player chooses a mixed strategy equivalent to a pure strategy.
If Player 1 chooses a mixed strategy $e_i = (0, ..., 0, 1, 0, ..., 0)$ with the entry 1 being the ith component and Player 2 chooses the mixed strategy Y, the expected payoff of Player 1 is

$$E(e_i, Y) = {}_iAY^T = \sum_{j=1}^{m} a_{ij} y_j. \tag{1.4}$$

Similarly, if Player 2 chooses the mixed strategy $e_j = (0, ..., 0, 1, 0, ..., 0)$ with the entry 1 being the jth component and Player 1 chooses the mixed strategy X, the expected payoff of Player 1 is

$$E(X, e_j) = XA_j = \sum_{i=1}^{n} x_i a_{ij}. \tag{1.5}$$

(ii) Both players choose mixed strategies equivalent to pure strategies.

If Player 1 chooses the mixed strategy $e_i = (0, ..., 0, 1, 0, ..., 0)$ with the entry 1 being the ith component and Player 2 chooses the mixed strategy $e_j = (0, ..., 0, 1, 0, ..., 0)$ with the entry 1 being the jth component, the expected payoff of Player 1 is

$$E(e_i, e_j) = a_{ij}. \tag{1.6}$$

The next lemma will be useful when calculating expected gains throughout the next chapters. In words, the lemma states that all pure strategies against a mixed strategy is as good as a mixed strategy against a mixed strategy. This means that if an inequality holds for the expected gain of Player 1 with a mixed strategy X no matter the pure strategy Player 2 chooses, then the result will hold even if Player 2 uses an arbitrary mixed strategy. Due to its importance in the next chapters and to the shortness of the argument, we also include the proof from [1].

Lemma 1.14. *Let $X \in S_n$ be a mixed strategy for Player 1 and α be a number such that $E(X, e_j) \geq \alpha$ for all j, then we also have $E(X, Y) \geq \alpha$ for any $Y \in S_m$.*

Proof. $E(X, e_j) \geq \alpha \Rightarrow \sum_{i=1}^{n} x_i a_{ij} \geq \alpha$. If we multiply both sides by $y_j \geq 0$ and sum on j, we have:

$$E(X, Y) = \sum_j \sum_i x_i a_{ij} y_j \geq \sum_j \alpha y_j = \alpha \tag{1.7}$$

since $\sum_j y_j = 1$. □

Example 1.15. Consider the mixed strategy $X = \left(\frac{2}{3}, \frac{1}{3}\right)$ for Player 1 in the game Matching Pennies described in Example 1.3. If Player 2 chooses Heads, the expected gain of Player 1 is:

$$E(X, e_1) = \sum_{i=1}^{2} x_i a_{i1} = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = 1/3$$

If Player 2 chooses TAILS, the expected gain of Player 1 is:

$$E(X, e_2) = \sum_{i=1}^{2} x_i a_{i2} = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = -1/3$$

Thus, the expected gain of Player 1 is at least $-\frac{1}{3}$ no matter the choice of pure strategy for Player 2 and therefore, Player 1 will have an expected gain of at least $-\frac{1}{3}$ no matter the mixed strategy chosen by Player 2.
In Example 1.12, we saw that when Player 2 chooses the mixed strategy $Y = \left(\frac{3}{4}, \frac{1}{4}\right)$, the expected gain of Player 1 is $\frac{1}{6}$ which is greater than $-\frac{1}{3}$.

1.2.2 Nash Equilibria

Other than the obvious interest in finding the winner, finding a stable solution in which each player is satisfied is an important part of the analysis of a game. This stable solution is called the Nash equilibrium, defined in words as follows:

Definition 1.16. A **Nash equilibrium** for a game with two or more players is a set of chosen strategies and the corresponding payoffs for which every player knows the equilibrium strategies of the other players and no player can increase his personal gain by changing his strategy when the others keeps theirs unchanged.

Let us now work towards giving a mathematical definition of this concept when considering mixed strategies. In a game, the goal of each of the players is to maximize their personal expected payoff. A saddle point in mixed strategies is a pair of strategies in a zero-sum game for which the payoffs to the players are optimal.

Definition 1.17. A **saddle point in mixed strategies** is a pair of mixed strategies (X^*, Y^*) that satisfies

$$E(X, Y^*) \leq E(X^*, Y^*) \leq E(X^*, Y) \text{ for all } X \in S_n, Y \in S_m. \tag{1.8}$$

The existence of a saddle point may or may not exist in every case with pure strategies. With mixed strategies, it is a known result from von Neumann's Minimax Theorem that a saddle point always exists.

Theorem 1.18. *For any $n \times m$ matrix A,*

$$\min_{Y \in S_m} \max_{X \in S_n} XAY^T = \max_{X \in S_n} \min_{Y \in S_m} XAY^T \tag{1.9}$$

*and this value is called the **value of the game**, $value(A)$. Moreover, there always exists at least one saddle point (X^*, Y^*), $X^* \in S_n, Y^* \in S_m$, such that*

$$E(X, Y^*) \leq E(X^*, Y^*) = value(A) \leq E(X^*, Y) \textit{ for all } X \in S_n, Y \in S_m. \tag{1.10}$$

Proof. For proof, see [3]. □

Example 1.19. The game Matching Pennies described in Example 1.3 is a zero-sum game. (X^*, Y^*) where $X^* = (\frac{1}{2}, \frac{1}{2})$ and $Y^* = (\frac{1}{2}, \frac{1}{2})$ is a saddle point in mixed strategies and the value of the game is 0.

When the game considered is not zero-sum, there are two expressions for the expected gain: $E_1(X, Y)$ and $E_2(X, Y)$ are respectively the expected payoffs of Player 1 and Player 2 when Player 1 chooses the mixed strategy X and Player 2 chooses the mixed strategy Y. In this case, the concept of optimal play is due to John Nash, so we have the concept of a Nash equilibrium.

Definition 1.20. A pair of mixed strategies (X^*, Y^*), $X^* \in S_n$, $Y^* \in S_m$ is a **Nash equilibrium** if

$$\begin{aligned} E_1(X, Y^*) &\leq E_1(X^*, Y^*), \text{ for all } X \in S_n \\ \text{and } E_2(X^*, Y) &\leq E_2(X^*, Y^*), \text{ for all } Y \in S_m. \end{aligned} \tag{1.11}$$

If (X^*, Y^*) is a Nash equilibrium, then $E_1(X^*, Y^*)$ and $E_2(X^*, Y^*)$ are the optimal payoffs to Player 1 and Player 2 respectively.

In matrix form, (X^*, Y^*), $X^* \in S_n$, $Y^* \in S_m$ is a **Nash equilibrium** if

$$\begin{aligned} E_1(X^*, Y^*) = X^*AY^{*^T} &\geq XAY^{*^T} \text{ for every } X \in S_n \\ \text{and } E_2(X^*, Y^*) = X^*AY^{*^T} &\geq X^*AY^T \text{ for every } Y \in S_m. \end{aligned} \tag{1.12}$$

As with the saddle points, a Nash equilibrium may or may not exist in every case with pure strategies, but the Nash Theorem of game theory states that if we allow mixed strategies, all games with a finite number of pure strategies has a Nash equilibrium (see [23],[24]).

Example 1.21. Consider a modification to the payoffs in the game Matching Pennies described in Example 1.3. Suppose the game matrices of Player 1 and Player 2 are

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \text{ and } B = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}.$$

This version of the game is not zero-sum. $X^* = (\frac{3}{5}, \frac{2}{5})$ and $Y^* = (\frac{2}{5}, \frac{3}{5})$ is a mixed Nash equilibrium where the optimal payoffs to Player 1 and Player 2 are $E_1(X^*, Y^*) = \frac{1}{5}$ and $E_2(X^*, Y^*) = \frac{1}{5}$.

1.2.3 Safe Games

In a two-player zero-sum game, whatever one player wins, the other player loses. Thus, while maximizing his personal payoff, a player also minimizes the payoff to his opponent. Hence, players want to choose strategies that will maximize their individual payoffs. In the case of non zero-sum games, this is not necessarily the case and so choosing a strategy is more complex. On that account, the concept of safe game, where players consider worst-case scenarios associated to their strategies, becomes interesting. When adopting a safe strategy in a safe game, a player asks himself what is the worst payoff he could get using this strategy; he finds his payoff in the situation where his opponent happens to choose the strategy which causes the greatest harm. In other words, the payoff that Player 1 can be guaranteed to receive is obtained by assuming that Player 2 is actually trying to minimize Player 1's payoff rather than maximizing his own. This leads to a minimal expected payoff, a useful information in a game where there is uncertainty about the other players. In real life situations, like the competitive diffusion of rival companies, being unsure of our opponents is likely to be the case.

Consider the two-player bimatrix game (A, B) where A is the game matrix of Player 1 and B is the game matrix of Player 2. To get the optimal guaranteed gains for the players, the games from the two matrices are considered separately. Matrix A is considered as the matrix for a zero-sum game with Player 1 against Player 2 where Player 1 (the row player) wants to maximize her personal gain, while Player 2 (the column player) wants to minimize the payoff to Player 1. In a zero-sum game, recall that the value of the game is the optimal payoffs to the players. Thus, the safety value of Player 1 is $value(A)$. Similarly, matrix B^T is considered as the matrix of a zero-sum game of Player 2 (the row player) against Player 1 (the column player), so Player 2's safety value is $value(B^T)$. More formerly, we have:

Definition 1.22. In the bimatrix game (A, B), the **safety value** for Player 1 is $value(A)$ and the safety value for Player 2 is $value(B^T)$.
Furthermore, if A has the saddle point (X^A, Y^A), then X^A is called the **maxmin strategy for Player 1** and Y^A is called the **minmax strategy for Player 2**. Similarly, if B^T has the saddle point (X^{B^T}, Y^{B^T}), then X^{B^T} is the **maxmin strategy for Player 2** and Y^{B^T} is called the **minmax strategy for Player 1**.

Remark 1.23. If (X^*, Y^*) is a Nash equilibrium for the bimatrix game (A, B), then $E_1(X^*, Y^*) = X^*AY^{*^T} \geq value(A)$ and $E_2(X^*, Y^*) = X^*AY^{*^T} \geq value(B^T)$.

Example 1.24. Consider the variation of the game Matching Pennies in Example 1.21 where the game matrix of Player 1 is

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}.$$

Let us consider the safe game in the point of view of Player 1, i.e. let us consider the zero-sum game of the matrix A, where Player 1 wants to maximize her personal gain, while Player 2 wants to minimize the payoff to Player 1. Let $X = (x, 1 - x)$ be a mixed strategy for Player 1. From Lemma 1.14, we know that Player 1's expected gain against any mixed strategy Y of Player 2 is greater than or equal to the minimum of her gain over all the pure strategies for Player 2. If Player 2 chooses Heads, the expected gain of Player 1 is $2x - (1 - x) = 3x - 1$. If Player 2 chooses Tails the expected gain of Player 1 is $-x + (1 - x) = 1 - 2x$.

The intersection point of the two equations represented in the graph of Figure 1.1, $\max_x \min\{1-2x, 3x-1\}$, is at $x = \frac{2}{5}$. Thus, Player 1's maxmin strategy is $X = (\frac{2}{5}, \frac{3}{5})$ and her safety value is $\frac{1}{5}$.

1.3 Notions in Linear Programming

In the presence of a zero-sum game, linear programming is one approach to solving game matrices. In the last section, we saw that getting the safety value and optimal safe strategies of the players in the safe game of a bimatrix game (A, B), involved

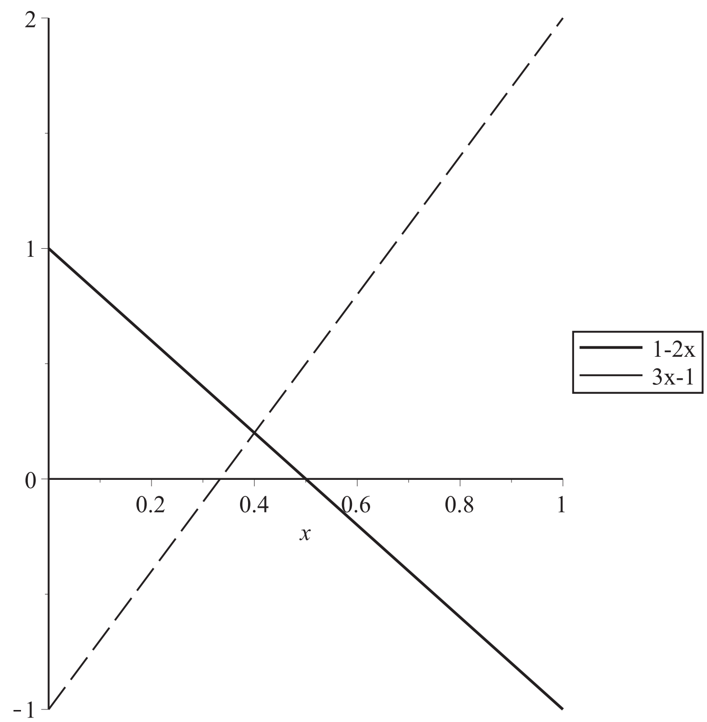


Figure 1.1: Expected gains of Player 1 in Example 1.24 when Player 2 chooses Heads or Tails.

separately solving the matrices as zero-sum games, regardless of the original game being zero-sum or not. Hence, the methods of linear programming will be useful to provide solutions to specific examples of the safe game of Competitive Diffusion.

Linear programming is a mathematical method for optimizing a linear objective function subject to constraints defined as linear equalities or inequalities. The following sections will define a linear programming problem and explain how the safe game can be solved with linear programming. Further details and examples of linear programming problems along with applications in game theory can be found in various books (e.g. [3],[33]).

1.3.1 Definition of a Linear Programming Problem

A linear programming problem is an optimization of a linear objective function subject to constraints defined as linear equalities or inequalities. Mathematically, we have:

Definition 1.25. A **linear programming problem** is defined as

$$\begin{aligned} &\text{maximize } \boldsymbol{c}^T\boldsymbol{X} \\ &\text{subject to } A\boldsymbol{X} \leq \boldsymbol{b} \\ &\boldsymbol{X} \geq \boldsymbol{0} \end{aligned} \tag{1.13}$$

where $\boldsymbol{X}$ is the column vector of variables to be determined, $\boldsymbol{c}$ and $\boldsymbol{b}$ are column vectors of coefficients known from the problem, A is a matrix of coefficients also known from the problem and $\boldsymbol{c}^T$ is the matrix transposition of $\boldsymbol{c}$.

The definition of the problem above is the **standard form** of a linear programming problem, where the goal is to maximize a function with constraints $\leq$ and where the variables of the vector $\boldsymbol{X}$ are positive.

Remark 1.26. Minimization problems, other forms of constraints as well as negative variables can all be rewritten in standard form. For minimization problems, it suffices to consider the maximization of the negative of the objective function. A linear equality constraint, $\boldsymbol{aX} = \boldsymbol{b}$, is equivalent to the pair of inequalities $\boldsymbol{aX} \leq \boldsymbol{b}$ and $-\boldsymbol{aX} \leq -\boldsymbol{b}$ and constraints with inequalities $\geq$ can be switched by taking the negative of the coefficients. Lastly, when a variable x is not restricted to be positive, one can introduce two new variables, x' and x'' such that $x' \geq 0$, $x'' \geq 0$ and replace x by $x' - x''$.

Every linear programming problem as a primal problem has a dual problem.

Definition 1.27. The dual problem of the linear programming problem from Definition 1.25 is defined as follows:

$$\begin{aligned} &\text{minimize } \boldsymbol{b}^T \boldsymbol{Y} \\ &\text{subject to } A^T \boldsymbol{Y} \geq \boldsymbol{c} \\ &\boldsymbol{Y} \geq \boldsymbol{0} \end{aligned} \tag{1.14}$$

where the column vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ as well as the matrix A are unchanged and $\boldsymbol{Y}$ is a new column vector of variables.

An important theorem concerning the primal and dual linear problems is the theorem of duality (see [33]).

Theorem 1.28 (Duality Theorem)**.** *If either the primal problem:*

$$\textit{Maximize } \mathbf{c}^T \boldsymbol{X}, \textit{ subject to } A\boldsymbol{X} \leq \boldsymbol{b} \textit{ and } \boldsymbol{X} \geq \mathbf{0}$$

or the dual problem:

$$\textit{Minimize } \boldsymbol{b}^T \boldsymbol{Y}, \textit{ subject to } A^T \boldsymbol{Y} \geq \boldsymbol{c} \textit{ and } \boldsymbol{Y} \geq \mathbf{0}$$

has a finite optimal solution, then so does the other problem and the objective functions are equal, i.e. $\max \mathbf{c}^T \boldsymbol{X} = \min \boldsymbol{b}^T \boldsymbol{Y}$.

The theorem of duality also indirectly states that any feasible solution to the dual problem is an upper bound to the optimal solution of the primal problem while any feasible solution to the primal problem is a lower bound to the optimal solution of the dual problem.

Note that it is possible for a linear programming problem to be unbounded or infeasible.

1.3.2 Solving the Safe Game with Linear Programming

We will now show how the problem of finding the safety value of Player 1 as well as her optimal safe strategy in a two-player game can be translated into a linear programming problem.

Suppose two players play a game where there are n pure strategies available to both players. Let $X = (x_1, ..., x_n)$ be a mixed strategy for Player 1 represented by a column vector, $\boldsymbol{X}$, and let $A = (a_{ij})$ be the game matrix of Player 1. If A^T is the transpose of the game matrix A, then $A^T\boldsymbol{X}$ is a vector of n entries with the jth entry, $\sum_{i=1}^{n} a_{ij}x_i$ corresponding to $E(X, e_j)$. From Lemma 1.14, we know that if $E(X, e_j) \geq \alpha$ for all j, then $E(X, Y) \geq \alpha$ for all $Y \in S_n$. Thus, with the strategy X, Player 1 can be assured an expected gain greater than or equal to $\min_j E(X, e_j)$. Therefore, finding the optimal safe strategy is finding the values of $\{x_i \mid 1 \leq i \leq n\}$ such that $\min_j E(X, e_j)$ is maximized. This can be translated to the following:

$$\begin{gathered} \text{maximize } v \\ \text{subject to } A^T\boldsymbol{X} \geq v \\ \text{with } x_1 + x_2 + ... + x_n = 1 \\ \text{and } x_1, x_2, ..., x_n \geq 0 \end{gathered} \tag{1.15}$$

where v is a variable corresponding to the minimal expected gain. If we introduce the column vectors $\boldsymbol{X'} = (x_1, x_2, ..., x_n, v)$ and $\boldsymbol{c} = (0, 0, ..., 0, 1)$, this problem is equivalent to

$$\text{maximize } \boldsymbol{c}^T\boldsymbol{X'}$$
$$\text{subject to}$$

$$\begin{array}{cccccccc} -a_{11}x_1 & -a_{21}x_2 & -... & -a_{n1}x_n & +v & \leq & 0 \\ -a_{12}x_1 & -a_{22}x_2 & -... & -a_{n2}x_n & +v & \leq & 0 \\ ... & ... & ... & ... & ... & & ... \\ -a_{1n}x_1 & -a_{2n}x_2 & -... & -a_{nn}x_n & +v & \leq & 0 \\ x_1 & +x_2 & +... & +x_n & +0v & = & 1 \end{array} \tag{1.16}$$

$$\text{and } x_1, x_2, ..., x_n \geq 0$$

which is a linear programming problem in standard form.

Moreover, we can derive the dual problem:

$$\text{minimize } \boldsymbol{b}^T\boldsymbol{Y'}$$
$$\text{subject to}$$
$$\begin{array}{cccccccc}
-a_{11}y_1 & -a_{12}y_2 & -... & -a_{1n}y_n & +V & \geq & 0 \\
-a_{21}y_1 & -a_{22}y_2 & -... & -a_{2n}y_n & +V & \geq & 0 \\
... & ... & ... & ... & ... & & ... \\
-a_{n1}y_1 & -a_{n2}y_2 & -... & -a_{nn}y_n & +V & \geq & 0 \\
y_1 & +y_2 & +... & +y_n & +0V & = & 1
\end{array} \tag{1.17}$$
$$\text{and } y_1, y_2, ..., y_n \geq 0$$

where $\boldsymbol{Y'} = (y_1, y_2, ..., y_n, V)$ and $\boldsymbol{b} = (0, 0, ..., 0, 1)$ are column vectors. If we introduce the column vector $\boldsymbol{Y} = (y_1, y_2, ..., y_n)$, this dual problem is equivalent to

$$\text{minimize } V$$
$$\text{subject to } A\boldsymbol{Y} \leq V \tag{1.18}$$
$$\text{with } y_1 + y_2 + ... + y_n = 1$$
$$\text{and } y_1, y_2, ..., y_n \geq 0$$

which translate to finding a mixed strategy for Player 2, $Y = (y_1, y_2, ..., y_n)$, such that the maximum gain that Player 1 can get considering all of her possible pure strategies is minimized. In other words, finding the optimal mixed strategy $Y = (y_1, y_2, ...y_n)$ for Player 2 such that $\max_i E(e_i, Y)$ is minimized.

From the theorem of duality, we know that if there exists a mixed strategy for Player 1 such that her minimal expected gain is maximized, then there also exists a mixed strategy for Player 2 such that the maximum expected gain of Player 1 is minimized. Moreover, these gains coincide and are precisely the safety value for Player 1 in the game. The safety value and the optimal safe strategy of Player 1 can be obtained by means of linear programming methods with the help of a mathematical programming language such as MATLAB® [19].

Furthermore, the theorem of duality indirectly states that any feasible solution of the primal is a lower bound of the dual and any feasible solution of the dual is an upper bound of the primal. This said, the minimal gain with any safe strategy for Player 1 is a lower bound on her safety value and Player 1's maximal gain against any mixed strategy for Player 2 is an upper bound. This will be a useful tool in the study of the safe game of Competitive Diffusion. While linear programming can provide the safety value and the optimal safe strategy of Player 1 for a two-player game, it requires the game matrix. Determining the game matrix of games can be a lot of work. Therefore, finding strategies for Player 1 and Player 2 that tightly bound the safety value is a better approach.

1.4 Notions in Graph Theory

Graph Theory is a branch of mathematics that studies the properties and applications of graphs.

Definition 1.29. A **graph** $G = (V, E)$ is a mathematical structure consisting of two finite sets V and E, their elements called **vertices** and **edges** respectively. Each edge is associated to one or two vertices, called its endpoint(s). Furthermore, we call a vertex in the graph joined by an edge to v a **neighbour** of v.

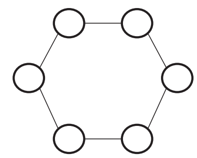
The structure of nodes and connections in graphs makes them useful models in various applications, including social networks. Social networks can be modelled by undirected graphs where the vertices correspond to the users of the network and edges represent links between two users. In fact, the game model of Competitive Diffusion, which will be described formally in the next chapter, is essentially a game on a graph. Thus, we will define a few families of graphs and some notions in graph theory that will be useful in the analysis of the game.

1.4.1 Common Graphs

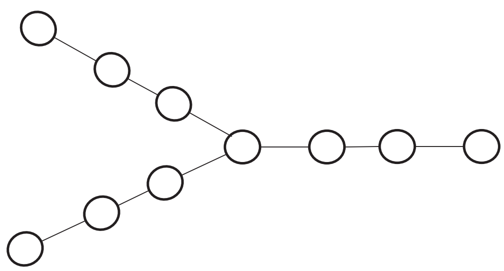
Definition 1.30. A **path** is a graph in which the number of vertices is one more than the number of edges and where the graph can be drawn so that all its vertices and edges lie on a straight line. A path graph with n vertices is denoted P_n.

Example 1.31. See Figure 1.2 (a).

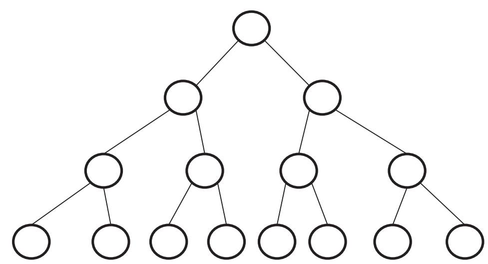
(a) Path graph with n vertices, P_n.



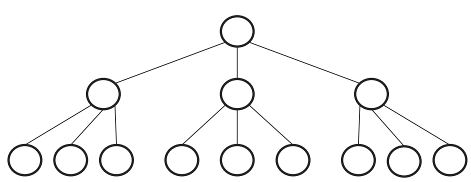
(b) Cycle graph with $n = 6$ vertices, C_6.



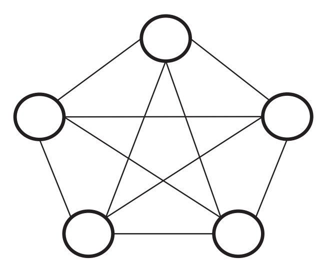
(c) Spider graph, S, with 10 vertices; 3 legs each having 3 vertices.



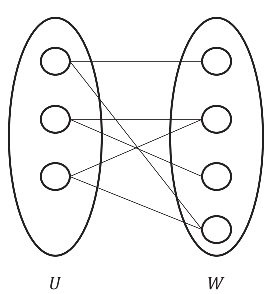
(d) Binary tree of height 3.



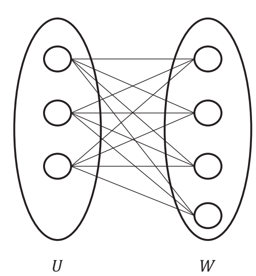
(e) Ternary tree of height 2.



(f) Complete graph, K_5.



(g) Bipartite graph.



(h) Complete bipartite graph.

Figure 1.2: Examples of common graphs.

Definition 1.32. A **cycle** is either a single vertex with a self-loop or a graph in which the number of vertices and edges are equal and the graph can be drawn so that all its vertices and edges lie on a single circle. A cycle graph with n vertices is denoted C_n.

Example 1.33. See Figure 1.2 (b).

A graph G is said to be **connected** if between every pair u and v of vertices of G, there exists an alternating sequence of vertices and edges from u to v in G. A **path** in a graph G is an alternating sequence of vertices and edges adjacent in G with no repeated vertices. A **cycle** in a graph G is an alternating sequence of vertices and edges adjacent in G that begins and ends with the same vertex but has no other repeated vertices.

Definition 1.34. A **tree** is a connected graph that has no cycle.

Definition 1.35. A **spider** with n vertices, S, is a tree with one and only one vertex of degree exceeding 2. The vertex of the spider with degree exceeding 2 is called the **body** of the spider and any branch at the body of the spider is called a **leg** of the spider. (Note that the legs of the spider are none other than non-trivial paths, see [26]).

Example 1.36. See Figure 1.2 (c).

Definition 1.37. A **rooted tree** is a tree with a designated vertex called the **root**. Each edge is considered to be directed away from the root.

A rooted tree imposes a hierarchy on the vertices according to their distance from the root. We can talk of the **depth** of a vertex v, that is, the distance from the root to v, and the **parent** and **children** of v, that is the vertex preceding v on the path from the root to v and the vertices succeeding v, respectively. Furthermore, the **height** of the tree is the length of the longest path from the root. Lastly, there are the **leaves** of the tree, particularly the vertices having no children or of degree 1 versus the **internal vertices** of the tree, which are the vertices that have at least one child.

Definition 1.38. An **m-ary tree** ($m \geq 2$) is a rooted tree in which every vertex has at most m children and in which at least one vertex has exactly m children.

Definition 1.39. A **complete m-ary tree** is an m-ary tree in which every internal vertex has exactly m children and all leaves have the same depth.

Example 1.40. See Figure 1.2 (d) and (e).

Definition 1.41. A **complete graph** is a graph in which every pair of vertices is joined by an edge. A complete graph on n vertices is denoted by K_n.

Example 1.42. See Figure 1.2 (f).

Definition 1.43. A **bipartite** graph is a graph G whose vertices V can be partitioned into two subsets U and W such that every edge of G has one endpoint in U and one endpoint in W.

Example 1.44. See Figure 1.2 (g).

Definition 1.45. A **complete bipartite** graph is a bipartite graph in which all the vertices in one of the bipartition subsets are joined to all the vertices of the other bipartition subset. A complete bipartite graph that has m vertices in one of its bipartite subsets and n vertices in the other bipartite subset is denoted $K_{m,n}$.

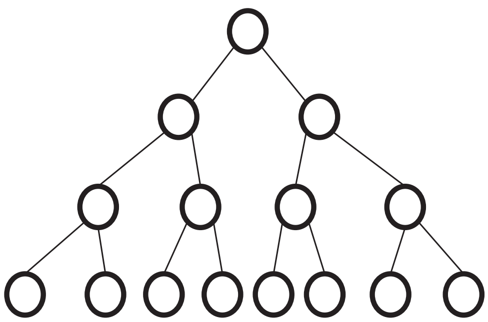
Example 1.46. See Figure 1.2 (h).

1.4.2 Properties of Trees

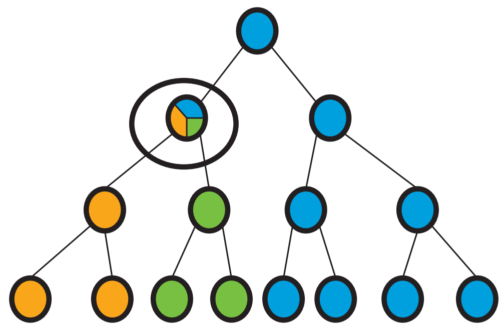
Definition 1.47. A **maximal sub-tree** of a graph G is a subgraph of G which is a tree and cannot be extended without creating a cycle.

Definition 1.48. A **branch** of a tree T_n at a vertex v is a maximal sub-tree of T_n which has v as a leaf. Correspondingly, the **weight** of the vertex v, $w(v)$, is the maximum number of edges in any branch of v. Furthermore, the **centroid** of T_n is the set of centroid vertices, vertices which have the minimal weight in T_n (See [22]).

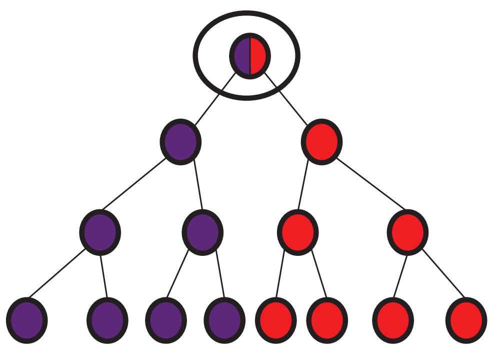
Example 1.49. Consider the tree with 15 vertices from Figure 1.3(a), T_{15} . In Figure 1.3(b), the vertex circled has three branches represented by the vertices coloured in Yellow, Green and Blue. The branch with Blue vertices has the maximum number of edges, 8. Thus, the weight assigned to the vertex circled is 8. Similarly, in Figure



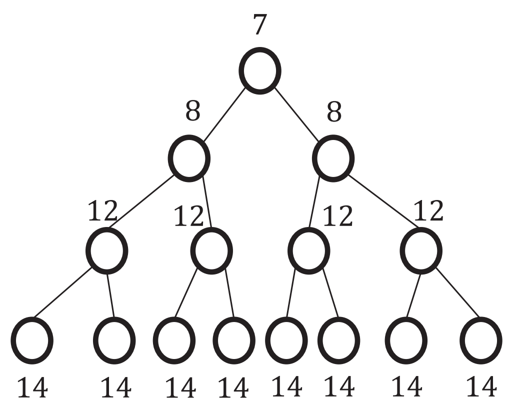
(a) Tree with 15 vertices, T_{15}.



(b) Vertex in T_{15} with weight 8.



(c) Vertex in T_{15} with weight 7.



(d) Weights in T_{15}.

Figure 1.3: Example of weights and centroid of a tree.

1.3(c), the vertex circled has two branches represented by the vertices coloured in Purple and Red. Both branches have 7 edges, thus this is the weight assigned to the vertex circled. In a similar manner, the weights can be assigned to the other vertices of T_{15} and we get the weights in Figure 1.3(d). The minimal weight in T_{15} is 7 and thus, the top vertex of T_{15} is the centroid since it is the only vertex with weight 7.

Theorem 1.50. *There are at most two centroid vertices in a tree and if two centroid vertices exist, they are adjacent. Moreover, a tree with two centroid vertices has an even number of vertices and the weight of each centroid vertex is $\frac{n}{2}$.*

Proof. See [15]. □

Theorem 1.51. *Let v be a vertex in T_n with k branches having $n_1, n_2, ..., n_k$ edges. Then, v is a centroid vertex of T_n if and only if $n_i \leq \frac{n}{2}$ for $1 \leq i \leq k$.*

Proof. See [13]. □

Definition 1.52. A **centroidal tree** is a tree which has only one vertex in its centroid. On the other hand, a **bicentroidal tree** is a tree which has two vertices in its centroid.

Chapter 2

Competitive Diffusion

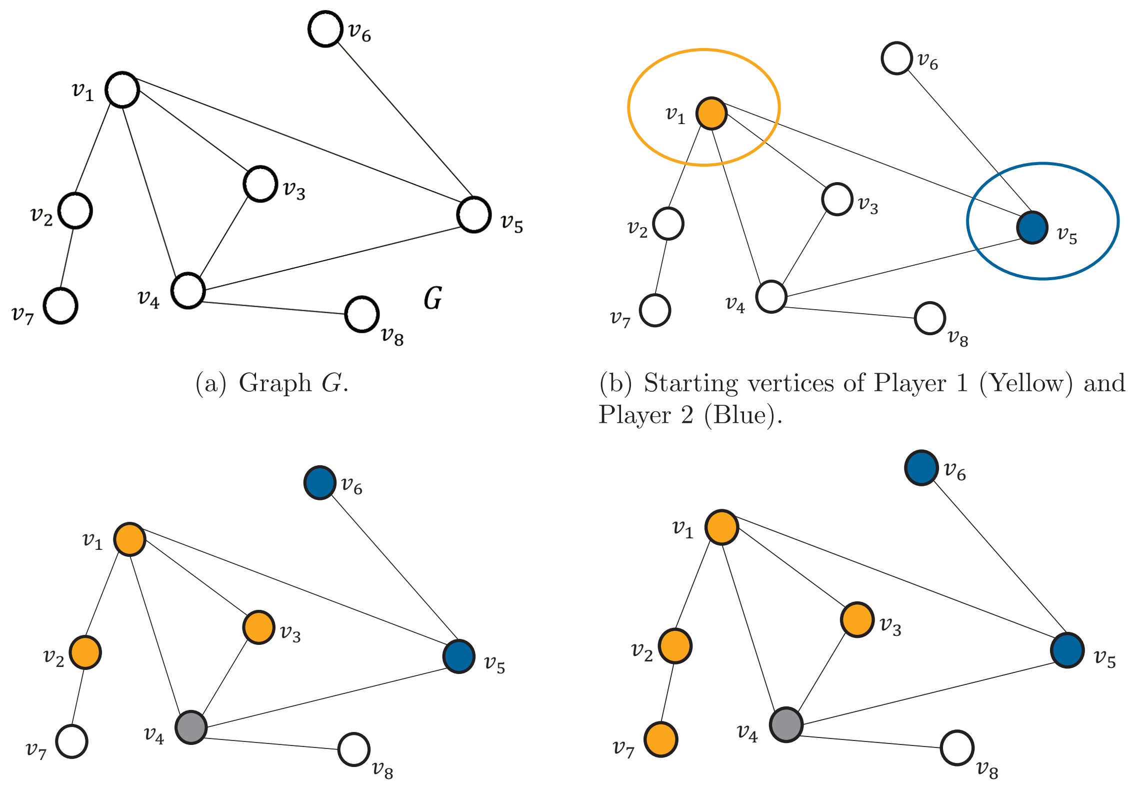
Now that we have defined the necessary notions, let us return our focus to Competitive Diffusion (see [1]), a game on a graph. Recall that Competitive Diffusion is a game-theoretic model for the diffusion of technologies, advertisement or other influences in social networks. It has players outside the network, like rival companies with a goal of reaching the most users.

2.1 Definition of the game

Let G be a graph on n vertices and suppose there are p players, $P_1, ..., P_p$ each having a distinct assigned colour (not white or grey). The strategy of each player is to choose a vertex in G as their starting vertex. The game begins by colouring each of the starting vertices of the players and then proceeding to the diffusion of the colours through G as follows: at each wave of diffusion, a vertex that has one or more neighbours with a certain colour inherits that colour while a vertex that has two neighbours with different colors turns grey. The grey concept translates the assumption that if the information about two companies reaches a user at the same time, they cancel each other. The diffusion finishes when all the vertices have either inherited a colour, have turned grey or are forced to stay white being blocked off by grey vertices. In the end, the gain of the players is the number of vertices with their assigned colour and clearly, the winner of the game is the player that has the greatest gain. We should note that if two or more players have the same starting vertex, then this vertex immediately turns grey.

While the game can be played with any finite number of players, this thesis will concentrate on the two-player version of the game. In the following, the two players will be called Player 1 (She) and Player 2 (He).

Example 2.1. Suppose two players, Player 1 and Player 2, play the game Competitive Diffusion on the graph G with 8 vertices in Figure 2.1(a). Let Player 1 have the



(a) Graph G. (b) Starting vertices of Player 1 (Yellow) and Player 2 (Blue). (c) After 1st wave of diffusion. (d) After 2nd wave of diffusion.

Figure 2.1: Example of a two-player Competitive Diffusion game.

assigned colour Yellow and Player 2 have the assigned colour Blue. Suppose Player 1 chooses the vertex v_1 and Player 2 chooses the vertex v_5 as shown in Figure 2.1(b). After the first wave of diffusion, the vertices that get coloured are represented in Figure 2.1(c). Note that the vertex v_4, a neighbour of v_1 and v_5 has turned grey. After the second wave of diffusion (see Figure 2.1(d)), the diffusion is done. All vertices have either inherited a colour, have turned grey or forced to stay white being blocked off by grey vertices (see for instance the vertex v_8 in Figure 2.1(d)). In the end, the gain of Player 1 is 4 vertices and the gain of Player 2 is 2 vertices, thus the winner is Player 1.

2.2 Strategies

We now introduce the terms and notations that will be used throughout the rest of this thesis for the strategies of the players. In the following, we suppose Player

1 and Player 2 play the game Competitive Diffusion on a graph G with n vertices $\{v_1, v_2, ..., v_n\}$.

The definition of the game states that the strategy of a player is to choose a starting vertex. These are the pure strategies of the players (see Definition 1.2). Hence, there are precisely n available pure strategies to the players when the game is played on the graph G, one for each possible starting vertex.

Recall from Definition 1.8 that a mixed strategy for a player is a vector with each element corresponding to the probability of the player choosing a specific pure strategy. For the game Competitive Diffusion, a mixed strategy for a player is a vector $X = (x_1, x_2, ..., x_n)$ where x_i equals the probability that the vertex v_i, $1 \leq i \leq n$, is chosen by the player. Correspondingly, we define the game matrix of Player 1 (see Definition 1.4) for the game Competitive Diffusion.

Definition 2.2. In the game Competitive Diffusion, the **game matrix of Player 1** on the graph G is denoted by A_G. Moreover, if $A_G = (a_{ij})$, then the entry a_{ij} of the matrix is the payoff to Player 1 when her strategy is to choose the vertex v_i and the strategy of Player 2 is to choose the vertex v_j.

Accordingly, the expected gain of Player 1 when she plays the mixed strategy X and Player 2 plays the mixed strategy Y is denoted by $Gain(G, X, Y)$ for the game Competitive Diffusion on the graph G. The expected gain, as in Definition 1.11, is equal to

$$Gain(G, X, Y) = XA_GY^T \tag{2.1}$$

where A_G is the game matrix of Player 1 on the graph G and Y^T is the transpose of the vector Y.

Now, when one or both players choose mixed strategies equivalent to pure strategies we will use the following notation.

Definition 2.3. Let v_k, $1 \leq k \leq n$, be a vertex in the graph G. The mixed strategy where a player chooses the vertex v_k with probability 1 and the other vertices with probability 0 is denoted by $Z(v_k)$. That is, $Z(v_k) = (z_1, z_2, ..., z_n)$ with

$$z_i = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

Thus, if u and v are two vertices of the graph G, then $Gain(G, Z(u), Z(v))$ is the expected gain of Player 1 when she chooses the vertex u and Player 2 chooses the vertex v.

2.3 Preliminary Results: Nash Equilibrium

Other than the obvious interest of finding the winner, finding a stable solution in which each player is satisfied is an important part of the analysis of a game. Pure Nash equilibria for the two-player game of Competitive Diffusion have been looked at for graphs including paths, cycles and trees by [5]. They can be summarized in the following results.

Theorem 2.4. *In a two-player game on a path with n vertices, the set of possible Nash equilibria is determined as below:*

(i) If n is even, then the two adjacent vertices in the middle form the only possible Nash equilibrium, and the equilibrium payoffs are equal and are $(n+1)/2$.

(i) If n is odd, then any two vertices in the middle (i.e., we have two possibilities, the central vertex and one of its neighbours) form a Nash equilibrium, and the equilibrium payoffs are equal and are $\lfloor n/2 \rfloor$.

Theorem 2.5. *In a two-player game on a tree T_n with centroid $C(T_n)$, the only possible pure Nash equilibria are determined as follows:*

(i) If $C(T_n) = \{c_1, c_2\}$, then $C(T_n)$ is the equilibrium and the equilibrium payoffs are equal to $\frac{n}{2}$.

(ii) If $C(T_n) = \{c\}$, then $\{c, v\}$ is the equilibrium where v is a neighbour of c on a branch at c with the maximum number of edges and the equilibrium payoffs are $n - w(v)$ and $w(v)$ respectively.

Theorem 2.6. *In a two-player game on a cycle C_n with n vertices, the set of possible Nash equilibria is determined as below:*

(i) If n is odd, then every two vertices on C_n, form a Nash equilibrium, and the payoffs are equal to $\frac{n-1}{2}$.

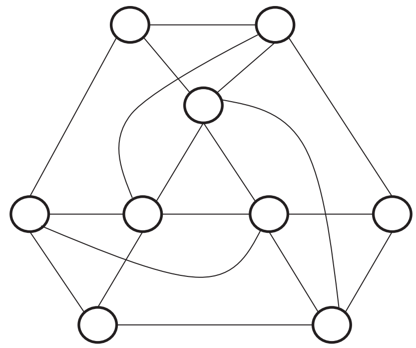


Figure 2.2: Example of a graph which does not admit a pure Nash equilibrium for the two-player game of Competitive Diffusion.

(ii) If n is even, then two vertices on C_n form a Nash equilibrium if and only if, they are of even distance, and the equilibrium payoffs are equal to $\frac{n}{2}$.

The gains in the pure Nash equilibrium will serve as comparison points for the safety value of the safe game. Despite these results, a pure Nash equilibrium does not necessarily exist for the two-player Competitive Diffusion on every graph. In [31], the authors present the graph of Figure 2.2 as an example for which the two-player game of Competitive Diffusion does not admit a pure Nash equilibrium.

2.4 Safe Game

The next chapters will concentrate on the safe game of the two-player Competitive Diffusion on various graphs. Recall from Section 1.2.3, that in the safe game a player asks himself what is the worst payoff he could get with a given strategy. Since the two players have the same strategy set, we will only consider the safe game in the point of view of Player 1. Thus, we consider the zero-sum game of Player 1 against Player 2 where Player 1 (the row player) wants to maximize her personal gain while Player 2 (the column player) wants to minimize the payoff to Player 1. In the physical sense of the game, this means that instead of caring just for the vertices of his colour, Player 2 cares for the total number of vertices that are his colour, grey or white. While it could happen that an opposing strategy for Player 2 would minimize Player 1's payoff and simultaneously give Player 2 the highest payoff, it does not need to be the case.

Thus, the safe game is not equivalent to the game where both players are trying to maximize their personal payoffs. The following is a simple example of this.

Example 2.7. Suppose Player 1 and Player 2 play the game Competitive Diffusion on a graph G with n vertices. Let u be a vertex in G. If the strategy of Player 1 is $Z(u)$, i.e. choosing the vertex u with probability 1, then the best strategy for Player 2, in the concept of the safe game, is to also adopt the strategy $Z(u)$ so that the payoff to Player 1 is minimized to 0. In this case, however, the payoff to Player 2 is also zero while if Player 2 chooses another vertex in G, his gain is at least 1. Thus, minimizing Player 1's payoff does not result in Player 2's highest payoff.

In the following definitions, we consider the two-player game of Competitive Diffusion on a graph G. Recall from Definition 1.22 that the safety value of Player 1 is the optimal guaranteed payoff of Player 1, or what is called the value of the game.

Definition 2.8. The **safety value for Player 1** in the game of Competitive Diffusion is

$$value(A_G) = \min_{Y \in S_n} \max_{X \in S_n} X A_G Y^T. \tag{2.3}$$

Definition 2.9. A **safe strategy** X for Player 1 is a mixed strategy of Player 1 adopted in a safe game setting. Correspondingly, the **guaranteed gain of Player 1** with the strategy X, $GGain(G, X)$, is the minimal gain that Player 1 could receive with the strategy X, i.e.

$$GGain(G, X) = \min_{Y \in S_n} X A_G Y^T. \tag{2.4}$$

From Lemma 1.14, we know that if an inequality on the expected gain holds for all pure strategies, then it will also hold no matter the mixed strategy. Thus, we also have the following:

$$GGain(G, X) = \min_{y} Gain(G, X, Z(y)) \tag{2.5}$$

where $y \in V(G)$ and $Z(y)$ is the vector as defined in Definition 2.3. In other words, the guaranteed gain of Player 1 with the strategy X on the graph G is the minimal payoff she can receive over all the possible starting vertices for Player 2 in G.

Definition 2.10. An **opposing strategy** Y for Player 2 is a mixed strategy for Player 2 in the safe game setting. Correspondingly, the **maximal gain of Player 1** against the strategy Y is the maximal gain that Player 1 could receive when Player 2 chooses the strategy Y, i.e.

$$MGain(G, Y) = \max_{X \in S_n} X A_G Y^T. \tag{2.6}$$

Again using the Lemma 1.14, we have

$$MGain(G, Y) = \max_{x} Gain(G, Z(x), Y) \tag{2.7}$$

where $x \in V(G)$ and $Z(x)$ is the vector as defined in Definition 2.3. In other words, the maximal gain of Player 1 against the opposing strategy Y of Player 2 on the graph G is the maximal gain she can receive over all of the possible starting vertices in G.

2.5 Linear Programming

As it was explained in Section 1.3.2, linear programming is a method that can solve the game matrix to get the safety value of Player 1 along with a maxmin strategy for Player 1 and a minmax strategy for Player 2. However, in order to use this method, the game matrix needs to be determined. In the game of Competitive Diffusion on a graph with n vertices, there are n pure strategies available to both players, thus n^2 entries in the game matrix. Determining all these payoffs is tedious work. Thus, linear programming will only be useful to solve a few specific examples. In the general case, we will use the fact that any feasible solution of the primal is a lower bound of the dual and any feasible solution of the dual is an upper bound on the primal. This being the case, the guaranteed gain with any safe strategy for Player 1 is a lower bound on the safety value while the maximal gain of Player 1 against a strategy of Player 2 is an upper bound on the safety value. Mathematically, we have the following:

$$\min_{y \in V(G)} Gain(G, X, Z(y)) \leq \min_{Y \in S_n} \max_{X \in S_n} X A_G Y^T \leq \max_{x \in V(G)} Gain(G, Z(x), Y) \tag{2.8}$$

which is equivalent to

$$GGain(G, X) \leq value(A_G) \leq MGain(G, Y). \tag{2.9}$$

where $value(A_G)$ is the safety value of Player 1 on the graph G and $GGain(G, X)$ and $MGain(G, Y)$ are respectively the guaranteed gain and maximal gain of Player 1 as defined in Definitions 2.9 and 2.10. In the following chapters, we will try to find strategies so that these bounds are tight. In other words, we want to find good safe strategies for Player 1 and good opposing strategies for Player 2.

Note: In the following chapters, the computations and simplifications were performed with the use of MATLAB® [19], Maple™, a trademark of Waterloo Maple Inc. [16] and Mathematica® [18].

Chapter 3

Paths and Spiders

In this chapter, we will study the two-player safe game of Competitive Diffusion on path graphs, P_n and spider graphs, S. Other than being interesting, the results of the study of special cases of trees, i.e. the paths and spiders in this chapter and the complete trees in the next chapter, will help gain insight into the safe game on trees in general in Chapter 5.

3.1 Paths

Recall from Definition 1.30 that a path graph with n vertices, P_n, is a graph that can be drawn so that all its vertices and edges lie on a straight line. Let the vertices of the path be labelled from v_1 to v_n, as shown in Figure 3.1. Recall from Definition 2.3, that the mixed strategy for which a player chooses a vertex v_k with probability 1 is denoted by $Z(v_k)$.

3.1.1 Game Matrix and Examples

Since the structure of P_n is simple, the game matrix of Player 1 in the two-player game can be determined.

Theorem 3.1. *In the two-player game of Competitive Diffusion on a path, P_n, let the strategy of Player 1 be choosing the vertex v_i, $1 \leq i \leq n$, and the strategy of Player 2 be choosing the vertex v_j, $1 \leq j \leq n$.*

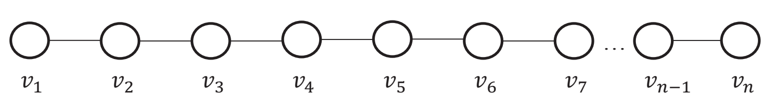


Figure 3.1: Labelling of the vertices on a path graph.

The game matrix of Player 1 is $A_{P_n} = (\pi_{ij}|_n)$ *where*

$$\pi_{ij}|_n = \begin{cases} \lfloor \frac{i+j-1}{2} \rfloor, & \text{if } i < j \\ 0, & \text{if } i = j \\ n - \lfloor \frac{i+j}{2} \rfloor, & \text{if } i > j. \end{cases} \tag{3.1}$$

Note: In the following, $\pi_{ij}|_n$ will represent the payoff of Player 1 when she chooses the vertex v_i and when Player 2 chooses the vertex v_j, $1 \leq i, j \leq n$ on a path with n vertices. When n is clear from the context, $\pi_{ij}|_n$ might be referred simply as π_{ij}. Moreover, by symmetry, this is also the game matrix for Player 2.

Proof. There are three cases to consider.

(i) When $i = j$, the two players have the same starting vertex. Thus, the vertex immediately turns grey and the gain of Player 1 is zero.

(ii) When $i < j$, the vertex v_i of Player 1 is to the left of the vertex v_j of Player 2 on the path. There are $i - 1$ vertices to the left of v_i and $j - i - 1$ vertices between the vertices v_i and v_j. If $j - i - 1$ is even, both players will gain exactly half of the vertices in between the starting vertices, i.e. $\frac{j-i-1}{2}$. If $j - i - 1$ is odd, each player will gain $\frac{j-i-2}{2}$ vertices and the vertex in the middle will turn grey. Thus, the total gain of Player 1 is $\left\lfloor \frac{i+j-1}{2} \right\rfloor$, the sum of the $i - 1$ vertices on the left of v_i, the vertex v_i and the $\left\lfloor \frac{j-i-1}{2} \right\rfloor$ vertices in between.

(iii) When $i > j$, the vertex v_i of Player 1 is more to the right than the vertex v_j of Player 2 on the path. There are $n - i$ vertices to the right of v_i and $i - j - 1$ vertices between the vertices v_i and v_j. Similarly to the case when $i < j$, the gain of Player 1 in between the starting vertices will be $\left\lfloor \frac{i-j-1}{2} \right\rfloor$. Thus, Player 1's total gain will be $n - \left\lfloor \frac{i+j}{2} \right\rfloor$, the sum of the $n - i$ vertices, the vertex v_i and the $\left\lfloor \frac{i-j-1}{2} \right\rfloor$ vertices in between.

□

With the game matrix in hand, we can use the methods of linear programming, as explained in Section 1.3.2, to determine the safety value of Player 1 along with the maxmin strategy of Player 1 and the minmax strategy of Player 2 on paths of specific lengths.

Example 3.2. On a path, P_n, with 12 vertices, the safety value of Player 1 is 4.4194. The maxmin strategy of Player 1 is:

v_i, i :	1	2	3	4	5	6	7	8	9	10	11	12
$X^A =$	(0.0000,	0.0000,	0.0000,	0.3226,	0.1129,	0.0645,	0.0645,	0.1129,	0.3226,	0.0000,	0.0000,	0.0000)

The minmax strategy of Player 2 is:

v_j, j :	1	2	3	4	5	6	7	8	9	10	11	12
$Y^A =$	(0.0000,	0.0000,	0.0000,	0.0269,	0.1720,	0.3011,	0.3011,	0.1720,	0.0269,	0.0000,	0.0000,	0.0000)

Example 3.3. On a path, P_n, with 13 vertices the safety value of Player 1 is 4.8387. The maxmin strategy of Player 1 is:

v_i, i :	1	2	3	4	5	6	7	8	9	10	11	12	13
$X^A =$	(0.0000,	0.0000,	0.0000,	0.0000,	0.2903,	0.1613,	0.0968,	0.1613,	0.12903,	0.0000,	0.0000,	0.0000,	0.0000)

The minmax strategy of Player 2 is:

v_j,j :	1	2	3	4	5	6	7	8	9	10	11	12	13
$Y^A =$	(0.0000,	0.0000,	0.0000,	0.0000,	0.1014,	0.2442,	0.3088,	0.2442,	0.1014,	0.0000,	0.0000,	0.0000,	0.0000)

As we can see from the examples, the probabilities in the maxmin and minmax strategies are symmetric with respect to the middle of the path and the positive probabilities are near the center of the path. We are looking to get an idea of the safety value of Player 1 on P_n as n gets large. This remark will motivate the definition of the strategy $C_P(k)$ for paths in the next section.

3.1.2 Bounds on the Safety Value with the Strategy $C_P(k)$

Motivated by the structure of the maxmin and minmax strategies of the examples in the last section, let us define a mixed strategy for a player on a path, P_n.

Definition 3.4. Let the **strategy** $C_P(k)$ be a mixed strategy where a player chooses a vertex from a set of central vertices with equal probability.

If n **is even**, the set consists of the $2k$ central vertices of the path, i.e. the strategy $C_P(k) = (z_1, z_2, ..., z_n)$ where

$$z_i = \begin{cases} 0, & \text{if } 1 \leq i < \frac{n}{2} - k + 1 \text{ or } \frac{n}{2} + k < i \leq n \\ \frac{1}{2k}, & \text{if } \frac{n}{2} - k + 1 \leq i \leq \frac{n}{2} + k \end{cases} \tag{3.2}$$

and $k \in \{1, 2, ..., \frac{n}{2}\}$.

If n **is odd**, the set consists of the $2k - 1$ central vertices of the path, i.e. the strategy $C_P(k) = (z_1, z_2, ..., z_n)$ where

$$z_i = \begin{cases} 0, & \text{if } 1 \leq i < \frac{n+1}{2} - k + 1 \text{ or } \frac{n+1}{2} + k - 1 < i \leq n \\ \frac{1}{2k-1}, & \text{if } \frac{n+1}{2} - k + 1 \leq i \leq \frac{n+1}{2} + k - 1 \end{cases} \tag{3.3}$$

and $k \in \{1, 2, ..., \frac{n}{2}\}$.

Recall from Section 2.5, that we have the following bound on the safety value of Player 1:

$$GGain(G, X) \leq \text{Safety Value of Player 1 on G} \leq MGain(G, Y) \tag{3.4}$$

where X is any safe strategy for Player 1 and Y is any opposing strategy for Player 2. Thus, we can consider the strategy $C_P(k)$ as a safe strategy for Player 1 and as an opposing strategy for Player 2 to get bounds on the safety value of Player 1 on a path. This leads to the following result.

Theorem 3.5. *In the two-player Competitive Diffusion game on P_n, the safety value of Player 1 is between $\frac{n}{2} - \frac{\sqrt{n}}{2} + \mathcal{O}(1)$ and $\frac{n}{2} - \frac{\sqrt{n}}{2\sqrt{3}} + \mathcal{O}(1)$ as $n \to \infty$.*

In order to prove this theorem, let us consider some lemmas on the guaranteed gain of Player 1 with the strategy $C_P(k)$ and the maximal gain of Player 1 when Player 2 has the strategy $C_P(k)$ as opposing strategy.

Lemma 3.6. *The guaranteed gain of Player 1 with the safe strategy $C_P(k)$ on a path with n vertices is*

(i) n even:

$$GGain(P_n, C_P(k)) = \frac{n}{2} - \frac{n}{4k} - \frac{k}{4} - \frac{1}{4} + \frac{1}{2k} \tag{3.5}$$

(ii) n odd:

$$GGain(P_n, C_P(k)) = \begin{cases} \frac{2kn-2n-k^2+2}{2(2k-1)}, & \textit{if } k \textit{ is even} \\ \frac{2kn-2n-k^2+1}{2(2k-1)}, & \textit{if } k \textit{ is odd.} \end{cases} \tag{3.6}$$

Proof. Recall from Definition 2.9, that the guaranteed gain of Player 1 with the mixed strategy $C_P(k)$ on the graph P_n is

$$GGain(P_n, C_P(k)) = \min_j Gain(P_n, C_P(k), Z(v_j)) \tag{3.7}$$

where $1 \le j \le n$. In other words, it is the minimal expected gain that Player 1 can get over all the possible starting vertices for Player 2. Since the gain of Player 1 and the strategy $C_P(k)$ are symmetric with respect to the middle of the path, we only need to consider the vertices on one half of the path as possible starting vertices for Player 2. Moreover, the expected gain of Player 1 when Player 2 chooses a vertex v_j with $1 \le j \le n$ is:

$$Gain(P_n, C_P(k), Z(v_j)) = \sum_{i=1}^{n} z_i \pi_{ij} \tag{3.8}$$

where π_{ij} is the entry of game matrix of Player 1 from Theorem 3.1 and z_i is the ith component of $C_P(k)$. Let us consider the cases when n is even and odd separately.

(i) n even:

The strategy $C_P(k)$ from Definition 3.4 for n even is $C_P(k) = (z_1, z_2, ..., z_n)$ where

$$z_i = \begin{cases} 0, & \text{if } 1 \le i < \frac{n}{2} - k + 1 \text{ or } \frac{n}{2} + k < i \le n \\ \frac{1}{2k}, & \text{if } \frac{n}{2} - k + 1 \le i \le \frac{n}{2} + k \end{cases} \tag{3.9}$$

and $k \in \{1, 2, ..., \frac{n}{2}\}$.

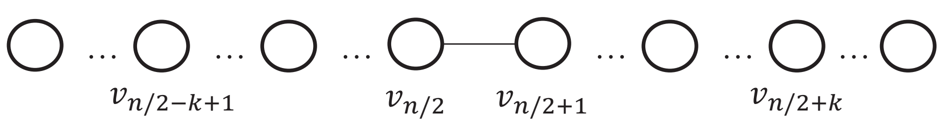


Figure 3.2: Illustration 1 in the proof of Lemma 3.6.

We can replace the values of z_i in (3.8) to get

$$Gain(P_n, C_P(k), Z(v_j)) = \frac{1}{2k} \sum_{i=\frac{n}{2}-k+1}^{\frac{n}{2}+k} \pi_{ij}. \tag{3.10}$$

Suppose the vertex v_j is on the left half of the path, i.e. $1 \leq j \leq \frac{n}{2}$.

(a) If $1 \leq j \leq \frac{n}{2} - k$ then $i > j$ for all $\frac{n}{2} - k + 1 \leq i \leq \frac{n}{2} + k$. Replacing the expression π_{ij} for $i > j$, we have

$$Gain(P_n, C_P(k), Z(v_j)) = \frac{1}{2k} \sum_{i=\frac{n}{2}-k+1}^{\frac{n}{2}+k} \left(n - \left\lfloor \frac{i+j}{2} \right\rfloor \right). \tag{3.11}$$

Hence, the expected gain of Player 1 decreases with j, and so the minimum if $1 \leq j \leq \frac{n}{2} - k$ is with $j = \frac{n}{2} - k$;

$$Gain\left(P_n, C_P(k), Z\left(v_{\frac{n}{2}-k}\right)\right) = \frac{1}{2k} \sum_{i=\frac{n}{2}-k+1}^{\frac{n}{2}+k} \left(n - \left\lfloor \frac{i + \frac{n}{2} - k}{2} \right\rfloor \right). \tag{3.12}$$

Expanding the summation, we get

$$\begin{aligned} Gain(P_n, C_P(k), Z\left(v_{\frac{n}{2}-k}\right)) = \frac{1}{2k} \Bigg(n - \left\lfloor \frac{n-2k+1}{2} \right\rfloor + n - \left\lfloor \frac{n-2k+2}{2} \right\rfloor \\ + n - \left\lfloor \frac{n-2k+3}{2} \right\rfloor + ... + n - \left\lfloor \frac{n-1}{2} \right\rfloor + n - \left\lfloor \frac{n}{2} \right\rfloor \Bigg). \end{aligned} \tag{3.13}$$

Since n is even, the floor functions can be reduced and we have

$$\begin{aligned} Gain(P_n, C_P(k), Z\left(v_{\frac{n}{2}-k}\right)) = \frac{1}{2k} \Bigg(n - \left(\frac{n}{2} - k\right) + n - \left(\frac{n}{2} - k + 1\right) \\ + n - \left(\frac{n}{2} - k + 1\right) + ... + n - \left(\frac{n}{2} - 1\right) + n - \left(\frac{n}{2}\right) \Bigg). \end{aligned} \tag{3.14}$$

This is equivalent to

$$Gain(P_n, C_P(k), Z\left(v_{\frac{n}{2}-k}\right)) = \frac{1}{2k} \left(2k\left(\frac{n}{2}\right) + 2\sum_{l=1}^{k-1} l + k \right). \tag{3.15}$$

Finally, we have

$$Gain(P_n, C_P(k), Z\left(v_{\frac{n}{2}-k}\right)) = \frac{n}{2} + \frac{k}{2} \tag{3.16}$$

since $\sum_{l=1}^{k-1} l = \frac{(k-1)k}{2}$.

(b) If $\frac{n}{2} - k + 1 \leq j \leq \frac{n}{2}$, then for $\frac{n}{2} - k + 1 \leq i \leq j - 1$, $i < j$ and for $j + 1 \leq i \leq \frac{n}{2} + k$, $i > j$. Replacing the expressions of π_{ij} in (3.10) gives

$$Gain(P_n, C_P(k), Z(v_j)) = \frac{1}{2k} \left[\sum_{i=\frac{n}{2}-k+1}^{j-1} \left\lfloor \frac{i+j-1}{2} \right\rfloor + \sum_{i=j+1}^{\frac{n}{2}+k} \left(n - \left\lfloor \frac{i+j}{2} \right\rfloor \right) \right]. \tag{3.17}$$

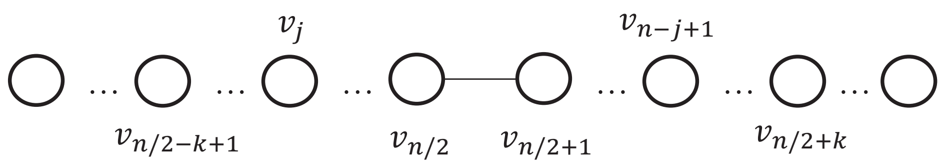


Figure 3.3: Illustration 2 in the proof of Lemma 3.6.

In order to simplify (3.17), let us determine the sum of the term $i = u$, $\frac{n}{2}-k+1 \leq u \leq j-1$ in the first summation with the term $i = n-u+1$ in the second summation:

$$\pi_{u,j} + \pi_{n-u+1,j} = \left\lfloor \frac{u+j-1}{2} \right\rfloor + n - \left\lfloor \frac{n-u+1+j}{2} \right\rfloor \tag{3.18}$$

where $\frac{n}{2} - k + 1 \leq u \leq j - 1$. To simplify this expression, we can use $\lfloor x + m \rfloor = \lfloor x \rfloor + m$ where m is an integer and consider the possible parities for u and j. We get

$$\pi_{u,j} + \pi_{n-u+1,j} = \frac{n}{2} + u - 1, \qquad \frac{n}{2} - k + 1 \leq u \leq j - 1. \tag{3.19}$$

Similarly, let us sum the terms $i = u$, $j + 1 \leq u \leq \frac{n}{2}$ and $i = n - u + 1$ in the second summation of (3.17),

$$\pi_{u,j} + \pi_{n-u+1,j} = 2n - \left\lfloor \frac{u+j}{2} \right\rfloor - \left\lfloor \frac{n-u+1+j}{2} \right\rfloor \tag{3.20}$$

where $j + 1 \leq u \leq \frac{n}{2}$. After simplification, we have

$$\pi_{u,j} + \pi_{n-u+1,j} = \frac{3n}{2} - j, \qquad j + 1 \leq u \leq \frac{n}{2}. \tag{3.21}$$

(b.1) When $j = \frac{n}{2} - k + 1$, replacing (3.21) in (3.17), gives

$$Gain\left(P_n, C_P(k), Z\left(v_{\frac{n}{2}-k+1}\right)\right) = \frac{1}{2k}\left[\sum_{i=\frac{n}{2}-k+2}^{\frac{n}{2}} \left(\frac{3n}{2} - j\right) + \left(n - \left\lfloor \frac{n+1}{2} \right\rfloor\right)\right] \tag{3.22}$$

which can be simplified to

$$Gain\left(P_n, C_P(k), Z\left(v_{\frac{n}{2}-k+1}\right)\right) = \frac{n}{2} - \frac{n}{4k} + \frac{1}{2k} + \frac{k}{2} - 1. \tag{3.23}$$

(b.2) When $\frac{n}{2} - k + 1 < j < \frac{n}{2}$, replacing (3.19) and (3.21) in (3.17), gives

$$Gain(P_n, C_P(k), Z(v_j)) = \frac{1}{2k}\left[\sum_{i=\frac{n}{2}-k+1}^{j-1}\left(\frac{n}{2}+i-1\right) + \sum_{i=j+1}^{\frac{n}{2}}\left(\frac{3n}{2}-j\right) + \left(n - \left\lfloor\frac{n-j+1+j}{2}\right\rfloor\right)\right] \tag{3.24}$$

which can be simplified to

$$Gain(P_n, C_P(k), Z(v_j)) = \frac{1}{2k}\left(\frac{-3jn}{2} - \frac{3j}{2} + 1 + \frac{3n^2}{8} + kn - \frac{k}{2} + \frac{3j^2}{2} - \frac{n}{4} - \frac{k^2}{2}\right) \tag{3.25}$$

where $\frac{n}{2}-k+1 < j < \frac{n}{2}$. We can determine that this expected gain is minimized with $j = \frac{n}{2} - 1$ and the corresponding gain is

$$Gain(P_n, C_P(k), Z(v_{\frac{n}{2}})) = \frac{n}{2} - \frac{n}{2k} + \frac{2}{k} - \frac{k}{4} - \frac{1}{4}. \tag{3.26}$$

(b.3) When $j = \frac{n}{2}$, replacing (3.19) in (3.17) gives

$$Gain\left(P_n, C_P(k), Z\left(v_{\frac{n}{2}-1}\right)\right) = \frac{1}{2k}\left[\sum_{i=\frac{n}{2}-k+1}^{\frac{n}{2}-1}\left(\frac{n}{2}+i-1\right) + \left(n - \left\lfloor\frac{n+1}{2}\right\rfloor\right)\right] \tag{3.27}$$

which can be simplified to

$$Gain\left(P_n, C_P(k), Z\left(v_{\frac{n}{2}}\right)\right) = \frac{n}{2} - \frac{n}{4k} + \frac{1}{2k} - \frac{k}{4} - \frac{1}{4}. \tag{3.28}$$

The expected gain from (3.28) is less than the expected gain when Player 2 chooses the vertex $v_{\frac{n}{2}-k}$ (see (3.16)), when Player 2 chooses the vertex $v_{\frac{n}{2}-k+1}$ (see (3.23)) and when Player 2 chooses the vertex $v_{\frac{n}{2}-1}$ (see (3.26)). Therefore, the minimal expected gain that Player 1 can get with the strategy $C_P(k)$ is when Player 2 chooses the vertex $v_{\frac{n}{2}}$ or symmetrically the vertex $v_{\frac{n}{2}+1}$. Thus, the guaranteed gain of Player 1 with the strategy $C_P(k)$ on P_n, n even is

$$GGain(P_n, C_P(k)) = \frac{n}{2} - \frac{n}{4k} + \frac{1}{2k} - \frac{k}{4} - \frac{1}{4}. \tag{3.29}$$

(ii) n odd:

The strategy $C_P(k)$ from Definition 3.4 for n odd is $C_P(k) = (z_1, z_2, ..., z_n)$ where

$$z_i = \begin{cases} 0, & \text{if } 1 \le i < \frac{n+1}{2} - k + 1 \text{ or } \frac{n+1}{2} + k - 1 < i \le n \\ \frac{1}{2k-1}, & \text{if } \frac{n+1}{2} - k + 1 \le i \le \frac{n+1}{2} + k - 1 \end{cases} \tag{3.30}$$

and $k \in \{1, 2, ..., \frac{n}{2}\}$.

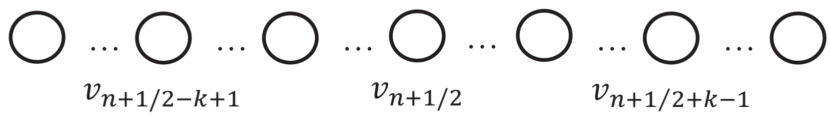


Figure 3.4: Illustration 3 in the proof of Lemma 3.6.

We can replace the values of z_i in (3.8) to get

$$Gain(P_n, C_P(k), Z(v_j)) = \frac{1}{2k-1} \sum_{i=\frac{n+1}{2}-k+1}^{\frac{n+1}{2}+k-1} \pi_{ij}. \tag{3.31}$$

Suppose the vertex v_j is on the left half of the path, i.e. $1 \le j \le \frac{n+1}{2}$. In a similar manner as with n even, we can determine the expected gain of Player 1 for the following cases:

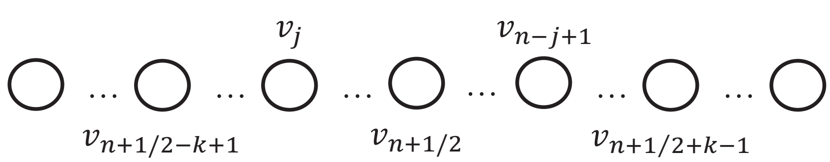


Figure 3.5: Illustration 4 in the proof of Lemma 3.6.

(a) $1 \le j \le \frac{n+1}{2} - k$

(b) $\frac{n+1}{2} - k + 1 \le j \le \frac{n-1}{2}$

(b.1) $j = \frac{n+1}{2} - k + 1$

(b.2) $\frac{n+1}{2} - k + 1 < j < \frac{n-1}{2}$

(b.3) $j = \frac{n-1}{2}$

(c) $j = \frac{n+1}{2}$.

We use the following formulas obtained by grouping the terms in pairs similarly as with n even:

$$\pi_{u,j} + \pi_{n-u+1,j} = \begin{cases} \frac{n-1}{2} + u - 1, & \text{if } u+j \text{ is even} \\ \frac{n-1}{2} + u, & \text{if } u+j \text{ is odd} \end{cases} \qquad \frac{n+1}{2} - k + 1 \leq u \leq j - 1 \tag{3.32}$$

and

$$\pi_{u,j} + \pi_{n-u+1,j} = \begin{cases} \frac{3n+1}{2} - j - 1, & \text{if } u+j \text{ is even} \\ \frac{3n+1}{2} - j, & \text{if } u+j \text{ is odd} \end{cases} \qquad j + 1 \leq u \leq \frac{n-1}{2}. \tag{3.33}$$

We determine that the minimum expected gain is case (c), i.e. when Player 2 chooses the vertex $v_{\frac{n+1}{2}}$ for which the expected gain of Player 1 is

$$GGain\left(P_n, C_P(k), Z\left(v_{\frac{n+1}{2}}\right)\right) = \begin{cases} \frac{2kn-2n-k^2+2}{2(2k-1)}, & \text{if } k \text{ is even} \\ \frac{2kn-2n-k^2+1}{2(2k-1)}, & \text{if } k \text{ is odd.} \end{cases} \tag{3.34}$$

Thus, the guaranteed gain of Player 1 with the strategy $C_P(k)$ on P_n, n odd is

$$GGain(P_n, C_P(k)) = \begin{cases} \frac{2kn-2n-k^2+2}{2(2k-1)}, & \text{if } k \text{ is even} \\ \frac{2kn-2n-k^2+1}{2(2k-1)}, & \text{if } k \text{ is odd.} \end{cases} \tag{3.35}$$

□

Now that we have an expression for the guaranteed gain of Player 1 with the strategy $C_P(k)$, we want to maximize her guaranteed gain over k.

Lemma 3.7. *The optimal guaranteed gain of Player 1 with the safe strategy $C_P(k)$ on P_n is $GGain(P_n, C_P(k^*)) = \frac{n}{2} - \frac{\sqrt{n}}{2} + \mathcal{O}(1)$ $(n \to \infty)$ where $k^* = \sqrt{n} + \mathcal{O}(1)$ $(n \to \infty)$ is the optimal integer k.*

Proof. We will prove the cases when n is even and odd separately.

(i) n even: From Lemma 3.6, we have the guaranteed gain of Player 1 with the strategy $C_P(k)$ on P_n, n even:

$$GGain(P_n, C_P(k)) = \frac{n}{2} - \frac{n}{4k} - \frac{k}{4} - \frac{1}{4} + \frac{1}{2k} \tag{3.36}$$

We want to maximize the guaranteed gain over k for n fixed. Let us define $f(x) = \frac{n}{2} - \frac{n}{4x} - \frac{x}{4} - \frac{1}{4} + \frac{1}{2x}$, so that $f(k) = GGain(P_n, C_P(k))$ whenever k is a positive integer. Using the derivative tests, we can maximize $f(x)$ over x in the interval $[1, \infty)$. $f(x)$ is a rational function and so is continuous at every point at which it is defined, i.e. at every point except $x = 0$. Thus, $f(x)$ is continuous on the interval $[1, \infty)$. Moreover, we have

$$\frac{df(x)}{dx} = \frac{n}{4x^2} - \frac{1}{2x^2} - \frac{1}{4} \tag{3.37}$$

and

$$\frac{df(x)}{dx} = 0 \Rightarrow x = -\sqrt{n-2}, \sqrt{n-2}. \tag{3.38}$$

Therefore, the only critical point in the interval $[1, \infty)$ is $x = \sqrt{n-2}$. Furthermore, we have $\frac{df(x)}{dx} > 0$ for $1 \leq x < \sqrt{n-2}$ and $\frac{df(x)}{dx} < 0$ for $\sqrt{n-2} < x$. Thus, $f(x)$ reaches a maximum at $x^* = \sqrt{n-2} = \sqrt{n} + \mathcal{O}(1)$. Hence, k^*, the integer maximizing $f(x)$, is either $\lfloor x^* \rfloor$ or $\lceil x^* \rceil$. Thus, $k^* \in [x^* - 1, x^* + 1]$ and so we have $k^* = \sqrt{n} + \mathcal{O}(1)$. Evaluating the function $f(x)$ at $x = k^*$ gives

$$\begin{aligned} f(k^*) = f(\sqrt{n} + \mathcal{O}(1)) = \frac{n}{2} - \frac{n}{4(\sqrt{n} + \mathcal{O}(1))} - \frac{(\sqrt{n} + \mathcal{O}(1))}{4} - \frac{1}{4} \\ + \frac{1}{2(\sqrt{n} + \mathcal{O}(1))} \end{aligned} \tag{3.39}$$

From the expansion of the Taylor series, we know that

$$(1 + x)^\alpha = 1 + \alpha x + \mathcal{O}(x^2) \tag{3.40}$$

for x small and $\alpha \in \mathbb{Q}$. Thus,

$$(\sqrt{n} + \mathcal{O}(1))^{-1} = \frac{1}{\sqrt{n}} \left(1 + \frac{\mathcal{O}(1)}{\sqrt{n}}\right)^{-1} = \frac{1}{\sqrt{n}} \left(1 - \frac{\mathcal{O}(1)}{\sqrt{n}} + \mathcal{O}\left(\frac{1}{n}\right)\right). \tag{3.41}$$

Using this to expand the terms of (3.39), we have

$$f(k^*) = f(\sqrt{n} + \mathcal{O}(1)) = \frac{n}{2} - \frac{\sqrt{n}}{2} + \mathcal{O}(1). \tag{3.42}$$

Furthermore, since $f(k) = GGain(P_n, C_P(k))$ whenever k is a positive integer, the optimal guaranteed gain is

$$GGain(P_n, C(k^*)) = \frac{n}{2} - \frac{\sqrt{n}}{2} + \mathcal{O}(1) \ (n \to \infty). \tag{3.43}$$

(ii) n odd: From Lemma 3.6, we have the guaranteed gain of Player 1 with the strategy $C_P(k)$ on P_n, n odd:

$$GGain(P_n, C_P(k)) = \begin{cases} \frac{2kn-2n-k^2+2}{2(2k-1)}, & \text{if } k \text{ is even} \\ \frac{2kn-2n-k^2+1}{2(2k-1)}, & \text{if } k \text{ is odd.} \end{cases} \tag{3.44}$$

We want to maximize the gain over k for n fixed. The proof is similar to the one for n even, defining the functions

$$g_1(x) = \frac{2xn - 2n - x^2 + 2}{2(2x-1)} \text{ and } g_2(x) = \frac{2xn - 2n - x^2 + 1}{2(2x-1)} \tag{3.45}$$

and considering that $g_1(x) = g_2(x) + \mathcal{O}(1)$ $(x \to \infty)$.

□

We now need a couple of lemmas on the maximal gain of Player 1 when Player 2 adopts the strategy $C_P(k)$ on P_n.

Lemma 3.8. *The maximal gain of Player 1 when Player 2 uses the opposing strategy $C_P(k)$ on a path with n vertices is:*

(i) n even:

$$MGain(P_n, C_P(k)) = \max\left\{\frac{n}{2} - \frac{k}{2} + \frac{1}{2}, \frac{n}{2} - \frac{n}{4k} + \frac{k}{4} - \frac{1}{4}\right\} \tag{3.46}$$

(ii) n odd:

$$MGain(P_n, C_P(k)) =$$

$$\max\left\{\frac{n}{2} - \frac{1}{2} - \frac{k^2}{2k-1} + \frac{k}{2k-1}, \begin{cases} \frac{2kn-2n-2k+k^2+2}{2(2k-1)}, & \textit{if } k \textit{ is even} \\ \frac{2kn-2n-2k+k^2+1}{2(2k-1)}, & \textit{if } k \textit{ is odd} \end{cases}\right\}. \tag{3.47}$$

Proof. Recall from Definition 2.10, that the maximal gain of Player 1 when Player 2 has the strategy $C_P(k)$ as opposing strategy is

$$MGain(P_n, C_P(k)) = \max_i Gain(P_n, Z(v_i), C_P(k)) \tag{3.48}$$

where $1 \le i \le n$. In other words, it is the maximal expected gain Player 1 can get over all her possible starting vertices. Again, since the gain of Player 1 on the path and the strategy $C_P(k)$ are symmetric with respect to the middle of the path, we only need to consider the vertices on one half of the path as possible starting vertices for Player 1. Moreover, the expected gain of Player 1 when she chooses a vertex v_i, $1 \leq i \leq n$ is

$$Gain(P_n, Z(v_i), C_P(k)) = \sum_{j=1}^{n} \pi_{ij} z_j \tag{3.49}$$

where π_{ij} are the entries of the game matrix of Player 1 from Theorem 3.1 and z_j is the jth element in the vector of the mixed strategy $C_P(k)$. Thus, this proof is similar to the proof of Lemma 3.6 except that we consider the possible starting vertices for Player 1 and that we are looking for the maximum expected gain. Let us consider the cases when n is even and odd separately.

(i) n even:

The strategy $C_P(k)$ from Definition 3.4 for n even is $C_P(k) = (z_1, z_2, ..., z_n)$ where

$$z_j = \begin{cases} 0, & \text{if } 1 \leq j < \frac{n}{2} - k + 1 \text{ or } \frac{n}{2} + k < j \leq n \\ \frac{1}{2k}, & \text{if } \frac{n}{2} - k + 1 \leq j \leq \frac{n}{2} + k \end{cases} \tag{3.50}$$

and $k \in \{1, 2, ..., \frac{n}{2}\}$.

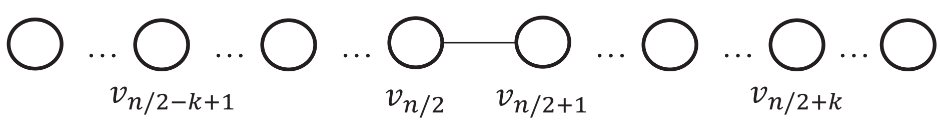


Figure 3.6: Illustration 1 in the proof of Lemma 3.8.

We can replace the values of z_j in (3.49) to get

$$Gain(P_n, Z(v_i), C_P(k)) = \frac{1}{2k} \sum_{j=\frac{n}{2}-k+1}^{\frac{n}{2}+k} \pi_{ij}. \tag{3.51}$$

Suppose the vertex v_i is on the left half of the path, i.e. $1 \leq i \leq \frac{n}{2}$. We determine the expected gain of Player 1 for the following cases:

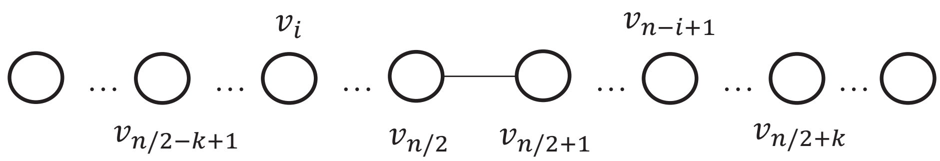


Figure 3.7: Illustration 2 in the proof of Lemma 3.8.

(a) $1 \leq i \leq \frac{n}{2} - k$

(b) $\frac{n}{2} - k + 1 \leq i \leq \frac{n}{2}$

(b.1) $i = \frac{n}{2} - k + 1$

(b.2) $\frac{n}{2} - k + 1 < i < \frac{n}{2}$

(b.3) $i = \frac{n}{2}$.

To simplify the expected gains, we use the following formulas obtained by grouping the terms in pairs similarly as in the proof of Lemma 3.6:

$$\pi_{i,u} + \pi_{i,n-u+1} = \frac{3n}{2} - u \qquad \frac{n}{2} - k + 1 \leq u \leq i - 1 \tag{3.52}$$

and

$$\pi_{i,u} + \pi_{i,n-u+1} = \frac{n}{2} + i - 1 \qquad i + 1 \leq u \leq \frac{n}{2}. \tag{3.53}$$

In the end, comparing the expected gains from the different cases, we determine that the maximal expected gain of Player 1 when Player 2 has the strategy $C_P(k)$ is the maximum of two expressions, one from case (a) (when she chooses the vertex $v_{\frac{n}{2}-k}$ or symmetrically, the vertex $v_{\frac{n}{2}+k+1}$) and the other from case (b.3) (when she chooses the vertex $v_{\frac{n}{2}}$ or symmetrically, the vertex $v_{\frac{n}{2}+1}$). Thus, the maximal expected gain of Player 1 is

$$MGain(P_n, C_P(k)) = \max\left\{\frac{n}{2} - \frac{k}{2} + \frac{1}{2}, \frac{n}{2} - \frac{n}{4k} + \frac{k}{4} - \frac{1}{4}\right\} \tag{3.54}$$

(ii) n odd:

The strategy $C_P(k)$ from Definition 3.4 for n odd is $C_P(k) = (z_1, z_2, ..., z_n)$ where

$$z_j = \begin{cases} 0, & \text{if } 1 \leq j < \frac{n+1}{2} - k + 1 \text{ or } \frac{n+1}{2} + k - 1 < j \leq n \\ \frac{1}{2k-1}, & \text{if } \frac{n+1}{2} - k + 1 \leq j \leq \frac{n+2}{2} + k - 1 \end{cases}. \tag{3.55}$$

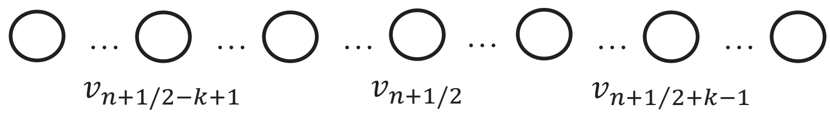


Figure 3.8: Illustration 3 in the proof of Lemma 3.8.

We can replace the values of z_j in (3.49) to get

$$Gain(P_n, Z(v_i), C_P(k)) = \frac{1}{2k-1} \sum_{j=\frac{n+1}{2}-k+1}^{\frac{n+1}{2}+k-1} \pi_{ij}. \tag{3.56}$$

Suppose the vertex v_i is on the left half of the path, i.e. $1 \leq i \leq \frac{n+1}{2}$. In a similar manner as with n even, we can determine the expected gain of Player 1 for the following cases:

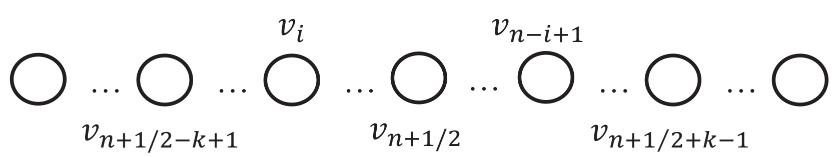


Figure 3.9: Illustration 4 in the proof of Lemma 3.8.

(a) $1 \leq i \leq \frac{n+1}{2} - k$

(b) $\frac{n+1}{2} - k + 1 \leq i \leq \frac{n-1}{2}$

- **(b.1)** $i = \frac{n+1}{2} - k + 1$
- **(b.2)** $\frac{n+1}{2} - k + 1 < i < \frac{n-1}{2}$
- **(b.3)** $i = \frac{n-1}{2}$

(c) $i = \frac{n+1}{2}$.

To simplify the expected gains, we use the following formulas obtained by grouping the terms in pairs:

$$\pi_{i,u}+\pi_{i,n-u+1} = \begin{cases} \frac{3n+1}{2} - u - 1, & \text{if } i+u \text{ is even} \\ \frac{3n+1}{2} - u, & \text{if } i+u \text{ is odd} \end{cases} \qquad \frac{n+1}{2}-k+1 \leq u \leq i-1 \tag{3.57}$$

and

$$\pi_{i,u} + \pi_{i,n-u+1} = \begin{cases} \frac{n-1}{2} + i - 1, & \text{if } i + u \text{ is even} \\ \frac{n-1}{2} + i, & \text{if } i + u \text{ is odd} \end{cases} \qquad i + 1 \leq u \leq \frac{n-1}{2}. \tag{3.58}$$

Similarly as for n even, the expected gain of Player 1 is the maximum of case (a) and (c), i.e. when she chooses the vertex $v_{\frac{n+1}{2}-k}$ (or symmetrically, the vertex $v_{\frac{n+1}{2}+k+1}$) and when she chooses the vertex $v_{\frac{n+1}{2}}$;

$$MGain(P_n, C_P(k)) =$$
$$\max\left\{\frac{n}{2} + \frac{k}{2k-1} - \frac{k^2}{2k-1} - \frac{1}{2}, \begin{cases} \frac{2kn-2n-2k+k^2+2}{2(2k-1)}, & \text{if } k \text{ is even} \\ \frac{2kn-2n-2k+k^2+1}{2(2k-1)}, & \text{if } k \text{ is odd} \end{cases}\right\}. \tag{3.59}$$

□

Now that we have an expression for the maximal gain of Player 1 against the strategy $C_P(k)$, Player 2 wants to minimize this maximal gain over k.

Lemma 3.9. *The optimal maximal gain of Player 1 when Player 2 uses the opposing strategy $C_P(k)$ on P_n with n even and n odd is $MGain(P_n, C(k^*)) = \frac{n}{2} - \frac{\sqrt{n}}{2\sqrt{3}} + \mathcal{O}(1)$ $(n \to \infty)$ where $k^* = \frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1)$ $(n \to \infty)$ is the optimal integer k.*

Proof. We will prove the cases when n is even and odd separately.

(i) n even: From Lemma 3.8 we have the maximal gain of Player 1 when Player 2 uses the strategy $C_P(k)$ on P_n, n even;

$$MGain(P_n, C_P(k)) = \max\left\{\frac{n}{2} - \frac{k}{2} + \frac{1}{2}, \frac{n}{2} - \frac{n}{4k} + \frac{k}{4} - \frac{1}{4}\right\}. \tag{3.60}$$

We want to minimize the maximal gain of Player 1 over k for n fixed. Let us define two functions, $f(x) = \frac{n}{2} - \frac{x}{2} + \frac{1}{2}$ and $g(x) = \frac{n}{2} - \frac{n}{4x} + \frac{x}{4} - \frac{1}{4}$ such that $\max\{f(k), g(k)\} = MGain(P_n, C_P(k))$ whenever k is a positive integer. The function $f(x)$ decreases with x and $g(x)$ increases with x. Thus, their intersecting point $x^* = \frac{1}{2} + \frac{\sqrt{9+12n}}{6} = \frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1)$ is the point x for which the maximum of the two functions $f(x)$ and $g(x)$ is minimized. In other words, the intersection point of the two functions is the $argmin_x\{\max\{f(x), g(x)\}\}$. Hence, the integer $k^* = argmin_k\{\max\{f(k), g(k)\}\}$, where k is an integer, is either $\lfloor x^* \rfloor$ or $\lceil x^* \rceil$. Thus, $k^* \in [x^* - 1, x^* + 1]$ and so we have $k^* = \frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1)$. Evaluating the functions $f(x)$ and $g(x)$ at $x = k^*$ gives

$$\begin{aligned} f(k^*) = f\left(\frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1)\right) &= \frac{n}{2} - \frac{\left(\frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1)\right)}{2} + \frac{1}{2} \\ &= \frac{n}{2} - \frac{\sqrt{n}}{2\sqrt{3}} + \mathcal{O}(1) \end{aligned} \tag{3.61}$$

and

$$\begin{aligned} g(k^*) = g\left(\frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1)\right) &= \frac{n}{2} - \frac{n}{4\left(\frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1)\right)} + \frac{\left(\frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1)\right)}{4} - \frac{1}{4} \\ &= \frac{n}{2} - \frac{\sqrt{n}}{2\sqrt{3}} + \mathcal{O}(1). \end{aligned} \tag{3.62}$$

Since $\max\{f(k), g(k)\} = MGain(P_n, C_P(k))$ whenever k is a positive integer, the maximal expected gain of Player 1 is

$$MGain(P_n, C(k^*)) = \frac{n}{2} - \frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1) \ (n \to \infty). \tag{3.63}$$

(ii) n odd: From Lemma 3.8 we have the maximal gain of Player 1 against Player 2 with the strategy $C_P(k)$ on P_n, n odd;

$$\begin{aligned} &MGain(P_n, C_P(k)) \\ &= \max\left\{\frac{n}{2} - \frac{1}{2} - \frac{k^2}{2k-1} + \frac{k}{2k-1}, \begin{cases} \frac{2kn-2n-2k+k^2+2}{2(2k-1)}, & \text{if } k \text{ is even} \\ \frac{2kn-2n-2k+k^2+1}{2(2k-1)}, & \text{if } k \text{ is odd} \end{cases}\right\}. \end{aligned} \tag{3.64}$$

We want to minimize the gain over k for n fixed. The proof is similar to the one for n even, defining the functions

$$f(x) = \frac{n}{2} - \frac{1}{2} - \frac{x^2}{2x-1} + \frac{x}{2x-1}, \tag{3.65}$$

$$g_1(x) = \frac{2xn - 2n - 2x + x^2 + 2}{2(2x-1)} \tag{3.66}$$

and

$$g_2(x) = \frac{2xn - 2n - 2x + x^2 + 1}{2(2x-1)}. \tag{3.67}$$

and considering that $g_1(x) = g_2(x) + \mathcal{O}(1)$ $(x \to \infty)$.

□

Now, we have the necessary results to prove Theorem 3.5.

Proof. (**Theorem 3.5**) Recall from (3.4),

$$GGain(G, X) \leq \text{Safety Value of Player 1 on G} \leq MGain(G, Y) \tag{3.68}$$

That is, the guaranteed gain of Player 1 with any safe strategy X is a lower bound on the safety value and the maximal gain of Player 1 against any opposing strategy Y for Player 2 is an upper bound on the safety value. In particular, the strategies X and Y can be the mixed strategy $C_P(k)$ and from Lemmas 3.7 and 3.9, we have

$$\begin{aligned} GGain(P_n, C_P(k), C_P(k)) = \frac{n}{2} - \frac{\sqrt{n}}{2} + \mathcal{O}(1) &\leq \text{Safety Value of Player 1 on } P_n \\ &\leq MGain(G, Y) = \frac{n}{2} - \frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1). \end{aligned} \tag{3.69}$$

□

Let us compare the bounds on the safety value with the payoffs in the pure Nash equilibrium for paths from Theorem 2.4.

Theorem 3.10. *For paths, P_n, the asymptotic value of the safety value for Player 1 approaches the best possible gain of any player in a Nash equilibrium, $\frac{n}{2}$ as n tends to infinity.*

Proof. This result directly follows from evaluating the limits of the bounds in Theorem 3.5 as $n \to \infty$. □

This is an interesting result since with a safe strategy, one expects his gain to be lower due to the fact that it is an assured expected gain. However, as we can see from the Theorem 3.10, we are assured an asymptotic gain as high as we could hope in any Nash equilibrium situation on a path.

	Lower bound on safety value $+\mathcal{O}(1)$	Proportion of n	**Upper bound on safety value** $+\mathcal{O}(1)$	Proportion of n	**Difference between bounds** $+\mathcal{O}(1)$	Proportion of n
n even						
10	3.42	0.342	4.09	0.409	0.67	0.067
20	7.76	0.388	8.71	0.436	0.95	0.048
50	21.46	0.429	22.96	0.459	1.49	0.030
100	45.00	0.450	47.11	0.471	2.11	0.021
1000	484.19	0.484	490.87	0.490	6.68	0.007
10000	4950.00	0.495	4971.13	0.497	21.13	0.002
n odd						
11	3.84	0.349	4.54	0.413	0.70	0.064
21	8.21	0.391	9.18	0.437	0.97	0.046
51	21.93	0.430	23.44	0.460	1.51	0.030
101	45.48	0.450	47.60	0.471	2.12	0.021
1001	484.68	0.484	491.37	0.491	6.69	0.007
10001	4950.50	0.495	4971.63	0.497	21.13	0.002

Table 3.1: Bounds on the safety value of Player 1 on P_n for specific values of n.

With this in mind, we could wonder if the asymptotic behaviour of the safety value is always related to the payoffs in the Nash equilibrium. However, we will see in the next chapter, that it is not true for all trees.

To finish the section on paths, Table 3.1 gives for specific values of n even and odd the bounds on the safety value of Player 1 calculated with the expressions in Theorem 3.5. We see that when n grows, the difference between the bounds as a proportion of n approaches 0 and the bounds themselves approach $\frac{n}{2}$.

3.2 Spiders

Recall from Definition 1.35 that a spider is a graph with one vertex of degree exceeding 2 called the body of the spider to which are attached non-trivial paths called the legs of the spider. Let us denote the m legs of a spider S by $\{s_1, s_2, ..., s_m\}$ and their lengths respectively by $\{l(s_1), l(s_2), ..., l(s_m)\}$. We will label a vertex v_i in S by an ordered pair (I_d, I_s) where I_d is the number of edges from the vertex v_i to the body of the spider and where $I_s \in \{s_1, s_2, ..., s_m\}$ is the leg the vertex belongs to. By convention, the body of the spider will be identified by the ordered pair $(0, 0)$ (see Figure 3.10). Since all the vertices in a leg of the spider are at a distinct distance from the body of the spider, their identification is well defined.

Furthermore, recall from Definition 2.3, that the mixed strategy for which a player chooses a vertex v_i with probability 1 is denoted by $Z(v_i)$. For spiders, it can equivalently be denoted by $Z((I_d, I_s))$ if the vertex v_i is identified by the ordered pair

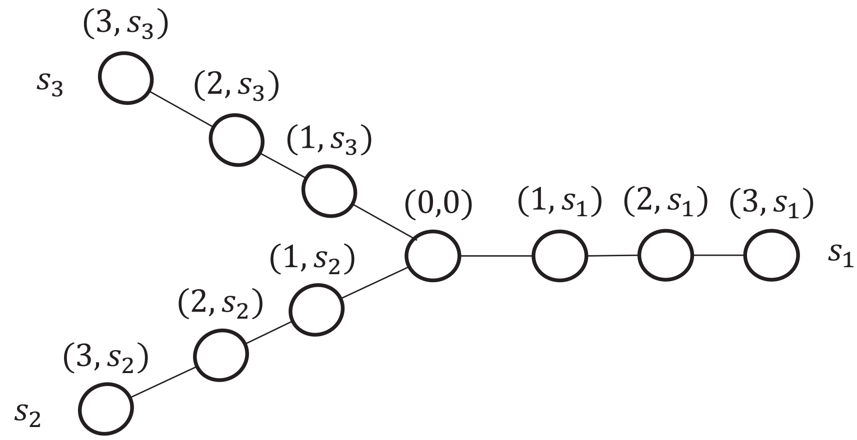


Figure 3.10: Labelling of the vertices in a spider with 3 legs.

(I_d, I_s). In order to simplify the notation, we will write $Z(I_d, I_s)$ instead of $Z((I_d, I_s))$ as it should not create any confusion.

3.2.1 Pure Nash Equilibrium

Since spiders are trees, we have the two-player pure Nash equilibrium from Theorem 2.5.

Proposition 3.11. *The centroid of a spider S with $m \geq 3$ legs each having l vertices is its body.*

Proof. Let b be the body of the spider S and consider the branches at b. There are m of them and they each have l vertices. Since, $l \leq \frac{n}{2}$ for $m \geq 3$ we have from Theorem 1.51 that b is a centroid vertex. Moreover, we know from Theorem 1.50 that if two centroid vertices exist in a tree, they are adjacent. Thus, the only possible other centroid vertex would be the first vertex on one of the legs. However, the first vertices of the legs have a branch with $n - l \geq \frac{n}{2}$ edges and thus are not centroid vertices by Theorem 1.51. Therefore, the centroid of S is its body. □

Corollary 3.12. *In the two-player game of Competitive Diffusion on a spider, S, with m legs each having l vertices, a pure Nash equilibrium is attained when one player has the body of the spider as starting vertex and the other player has a vertex* *adjacent to the body of the spider as starting vertex. Moreover, the payoffs of the players are respectively* $(m-1)l+1$ *and* l.

Proof. This result immediately follows from Theorem 2.5 having proved in Proposition 3.11 that the body of the spider is the centroid. □

3.2.2 Game Matrix and Examples

The game matrix of Player 1 in the two-player game on S can also be determined knowing the game matrix for paths (see Theorem 3.1).

Theorem 3.13. *In the two-player game of Competitive Diffusion on a spider,* S*, let the strategy of Player 1 be choosing a vertex* (I_d, I_s) *and the strategy of Player 2 be choosing a vertex* (J_d, J_s)*. The game matrix of Player 1 is* $A_S = (a_{(I_d,I_s),(J_d,J_s)})$ *where*

$$a_{(I_d,I_s),(J_d,J_s)} = \begin{cases} 0, & \text{if } I_s = J_s \text{ and } I_d = J_d \\ \pi_{I_d,J_d}\big|_{l(I_d)} + n - l(I_s), & \text{if } I_s = J_s \text{ and } I_d < J_d, \\ & \quad I_d, J_d \neq 0 \\ \pi_{I_d,J_d}\big|_{l(I_d)}, & \text{if } I_s = J_s \text{ and } I_d > J_d, \\ & \quad I_d, J_d \neq 0 \\ \pi_{l(I_s)-I_d+1,l(I_s)+J_d+1}\big|_{l(I_s)+l(J_s)+1} & \text{if } I_s \neq J_s \text{ and } I_d < J_d, \\ \quad +n - l(I_s) - l(J_s) - 1, & \quad I_d, J_d \neq 0 \\ \pi_{l(I_s)-I_d+1,l(I_s)+J_d+1}\big|_{l(I_s)+l(J_s)+1}, & \text{if } I_s \neq J_s \text{ and } I_d \geq J_d, \\ & \quad I_d, J_d \neq 0 \\ \pi_{1,J_d+1}\big|_{l(J_s)+1} + n - l(J_s) - 1, & \text{if } (I_s, I_d) = (0,0) \text{ and } J_d \neq 0 \\ \pi_{I_d+1,1}\big|_{l(I_s)+1}, & \text{if } (J_s, J_d) = (0,0) \text{ and } I_d \neq 0 \end{cases} \tag{3.70}$$

where $\pi_{i,j}\big|_n$ *are the entries in the game matrix of Player 1 on* P_n *as determined in Theorem 3.1.*

Proof. There are a few cases to consider.

(i) If $I_s = J_s$ and $I_d = J_d$, the two players have the same starting vertex since the vertices are uniquely identified by the ordered pairs. Thus, the gain of Player 1 is zero.

(ii) If $I_s = J_s$ and $I_d \neq J_d$, both non zero, the chosen vertices of Player 1 and Player 2 are on the same leg of S. In this case, the payoffs are the same as on a path having $l(I_s)$ vertices with Player 1 choosing the vertex v_{I_D} and Player 2 choosing the vertex v_{J_D} with the exception that the player claiming the vertex v_1 has an additional gain of $n - l(I_s)$ corresponding to the vertices in the other legs and the body of the spider. Thus,

$$a_{(I_d,I_s),(J_d,J_s)} = \begin{cases} \pi_{I_d,J_d}\big|_{l(I_d)} + n - l(I_s), & \text{if } I_d < J_d \\ \pi_{I_d,J_d}\big|_{l(I_s)}, & \text{if } I_d > J_d. \end{cases} \tag{3.71}$$

(iii) If $I_s \neq J_s$ and both are not zero, then the chosen vertices of Player 1 and Player 2 are on different legs of S. In this case, the payoffs are the same as on a path having $l(I_s) + l(J_s) + 1$ vertices with Player 1 choosing the vertex $v_{l(I_s)-I_d+1}$ and Player 2 choosing the vertex $v_{l(I_s)+J_d+1}$ with the exception that the player claiming the body of the spider has an additional gain of $n - l(I_s) - l(J_s) - 1$ corresponding to the vertices in the other legs of the spider. Thus,

$$a_{(I_d,I_s),(J_d,J_s)} = \begin{cases} \pi_{l(I_s)-I_d+1,l(I_s)+J_d+1}\big|_{l(I_s)+l(J_s)+1} & \\ \quad +n - l(I_s) - l(J_s) - 1, & \text{if } I_d < J_d \\ \pi_{l(I_s)-I_d+1,l(I_s)+J_d+1}\big|_{l(I_s)+l(J_s)+1}, & \text{if } I_d \geq J_d. \end{cases} \tag{3.72}$$

(iv) If $I_s \neq J_s$ and $I_s = 0$ or $J_s = 0$, then one of the players chooses the body of the spider and the other player chooses a vertex on one leg. If $J_s = 0$, the payoff to Player 1 is the same as on a path having $l(I_s) + 1$ vertices with Player 1 choosing the vertex v_{I_d+1} and Player 2 choosing the vertex v_1. If $I_s = 0$, the payoff to Player 1 is the same as on a path having $l(J_s) + 1$ vertices with Player 1 choosing the vertex v_1 and Player 2 choosing the vertex v_{J_d+1} plus $n - l(J_s) - 1$ vertices on the other legs. Thus,

$$a_{(I_d,I_s),(J_d,J_s)} = \begin{cases} \pi_{1,J_d+1}\big|_{l(J_s)+1} & \\ \quad +n - l(J_s) - 1, & \text{if } (I_s, I_d) = (0,0) \text{ and } J_d \neq 0 \\ \pi_{I_d+1,1}\big|_{l(I_s)+1}, & \text{if } (J_s, J_d) = (0,0) \text{ and } I_d \neq 0 \end{cases} . \tag{3.73}$$

□

Replacing the values for $\pi_{ij}|_n$ from Theorem 3.1 when the spider has legs with equal length, the expression of the game matrix can be simplified to the following.

Corollary 3.14. *In the two-player game of Competitive Diffusion on a spider, S, with m legs each having l vertices, let the strategy of Player 1 be choosing a vertex (I_d, I_s) and the strategy of Player 2 be choosing a vertex (J_d, J_s). The game matrix of Player 1 is $A_S = (a_{(I_d,I_s),(J_d,J_s)})$ where $a_{(I_d,I_s),(J_d,J_s)} =$*

$$\begin{cases} 0, & \text{if } I_s = J_s \text{ and } I_d = J_d \\ \left\lfloor \frac{I_d+J_d-1}{2} \right\rfloor + n - l, & \text{if } I_s = J_s \text{ and } I_d < J_d,\ I_d, I_d \neq 0 \\ l - \left\lfloor \frac{I_d+J_d}{2} \right\rfloor, & \text{if } I_s = J_s \text{ and } I_d > J_d,\ I_d, J_d \neq 0 \\ \left\lfloor \frac{-I_d+J_d+1}{2} \right\rfloor + n - l - 1, & \text{if } I_s \neq J_s \text{ and } I_d < J_d,\ I_d, J_d \neq 0 \\ l + \left\lfloor \frac{-I_d+J_d+1}{2} \right\rfloor, & \text{if } I_s \neq J_s \text{ and } I_d > J_d,\ I_d, J_d \neq 0 \\ l, & \text{if } I_s \neq J_s \text{ and } I_d = J_d,\ I_d, J_d \neq 0 \\ \left\lfloor \frac{1+J_d}{2} \right\rfloor + n - l - 1, & \text{if } (I_s, I_d) = (0,0) \text{ and } J_d \neq 0 \\ l - \left\lfloor \frac{I_d}{2} \right\rfloor, & \text{if } (J_s, J_d) = (0,0) \text{ and } I_d \neq 0. \end{cases} \tag{3.74}$$

Proof. This is simply the same expression as Theorem 3.13 with all the lengths of legs replaced by l and the corresponding expressions of $\pi_{ij}|_n$ from Theorem 3.1. □

With the game matrix in hand, we can again use the methods of linear programming as it was explained in Section 1.3.2 to determine the safety value of Player 1 along with the maxmin strategy of Player 1 and the minmax strategy of Player 2 on a spider with legs having the same number of vertices.

Example 3.15. On a spider S with 3 legs each having 10 vertices, the safety value of Player 1 is 9.1643. The maxmin strategy of Player 1 is $X^A = (x_1, x_2, ..., x_n)$ where x_i, the probability of choosing the vertex $v_i = (I_d, I_s)$ for $I_d \in \{1, 2, ..., 10\}$ and $I_s \in \{s_1, s_2, s_3\}$, is:

x_i :	Distance from the body of the spider : I_d										
I_s	0	1	2	3	4	5	6	7	8	9	10
Body of the spider	0.0060	-	-	-	-	-	-	-	-	-	-
s_1	-	0.0566	0.2766	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
s_2	-	0.0566	0.2766	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
s_3	-	0.0566	0.2766	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

The minmax strategy of Player 2 is $Y^A = (y_1, y_2, ..., y_n)$ where y_j, the probability of choosing the vertex $v_j = (J_d, J_s)$ for $J_d \in \{1, 2, ..., 10\}$ and $J_s \in \{s_1, s_2, s_3\}$, is:

y_j :	Distance from the body of the spider J_d										
J_s	0	1	2	3	4	5	6	7	8	9	10
Body of the spider	0.5636	-	-	-	-	-	-	-	-	-	-
s_1	-	0.1314	0.0141	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
s_2	-	0.1314	0.0141	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
s_3	-	0.1314	0.0141	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

As we can see from the example, the probabilities in the maxmin and minmax strategies are the same on each of the three legs and the positive probabilities are near the body of the spider.

3.2.3 Bounds on Safety Value with the Strategy $C_{S1}(k)$

Motivated by the structure of the maxmin and minmax strategies of Example 3.15, let us define a mixed strategy for a player on a spider, S, with m legs each having l vertices.

Definition 3.16. Let the **strategy** $C_{S1}(k)$ be a mixed strategy where a player chooses a vertex from a set of central vertices with equal probability. The set consists of the body of the spider and the first k vertices of the m legs, i.e. the strategy $C_{S1}(k) = (z_1, z_2, ..., z_n)$ where z_i, the probability of choosing the vertex $v_i = (I_d, I_s)$ where $I_d \in \{1, 2, ..., l\}$ and $I_s \in \{s_1, s_2, ..., s_m\}$, or $I_d = 0$ and $I_s = 0$ is:

$$z_i = \begin{cases} 0, & \text{if } k < I_d \leq l \\ \frac{1}{mk+1}, & \text{if } 0 \leq I_d \leq k \end{cases} \tag{3.75}$$

and $k \in \{0, 1, ..., l\}$.

In a similar manner to paths, we can consider the strategy $C_{S1}(k)$ as a safe strategy for Player 1 and as an opposing strategy for Player 2 to get bounds on the safety value of Player 1 on a spider with legs of equal length. This leads to the following result.

Theorem 3.17. *In the two-player Competitive Diffusion on S with m legs each having l vertices, the safety value of Player 1 is between $l - \frac{\sqrt{l}}{\sqrt{m}} + \mathcal{O}(1)$ $(l \to \infty)$ and l.*

In order to prove this theorem, let us consider some lemmas on the guaranteed gain of Player 1 with the safe strategy $C_{S1}(k)$ and the maximal gain of Player 1 against the strategy $C_{S1}(k)$.

Lemma 3.18. *The guaranteed gain of Player 1 with the safe strategy $C_{S1}(k)$ on a spider, S, with m legs each having l vertices is*

$$GGain(S, C_{S1}(k)) = \begin{cases} \frac{m}{mk+1}\left(kl - \frac{k^2}{4}\right), & \text{if } k \text{ is even} \\ \frac{m}{mk+1}\left(kl - \frac{k^2}{4} + \frac{1}{4}\right), & \text{if } k \text{ is odd.} \end{cases} \tag{3.76}$$

Proof. From Definition 2.9, we have the guaranteed gain of Player 1 with the mixed strategy $C_{S1}(k)$ on S;

$$GGain(S, C_{S1}(k)) = \min_j Gain(S, C_{S1}(k), Z(v_j)) \tag{3.77}$$

where $1 \leq j \leq n$. In other words, it is the minimal expected gain Player 1 can get over all the possible starting vertices of Player 2. Due to symmetry, we only need to consider the body of the spider and the vertices on one of the legs of S as possible starting vertices for Player 2. Moreover, the expected gain of Player 1 when Player 2 chooses a vertex $v_j = (J_d, J_s)$ is

$$Gain(S, C_{S1}(k), Z(J_d, J_s)) = \sum_{i=1}^{n} z_i a_{v_i,(J_d,J_s)} \tag{3.78}$$

where $a_{v_i,(J_d,J_s)}$ is equivalent to the entry of the game matrix of Player 1 from Corollary 3.14 when Player 1 chooses the vertex $v_i = (I_d, I_s)$ and Player 2 chooses the vertex (J_d, J_s), $a_{(I_d,I_s),(J_d,J_s)}$, and where z_i is the ith component of the strategy $C_{S_1}(k)$. We can replace the value of z_i from Definition 3.16 to get

$$Gain(S, C_{S1}(k), Z(J_d, J_s)) = \frac{1}{mk+1}\left[a_{(0,0),(J_d,J_s)} + \sum_{I_s=1}^{m}\sum_{I_d=1}^{k} a_{(I_d,I_s),(J_d,J_s)}\right]. \tag{3.79}$$

(a) If $v_j = (J_d, J_s)$ with $J_d > k$ and $J_s \neq 0$, replacing the expression of $a_{(I_d,I_s),(J_d,J_s)}$ from Corollary 3.14 in (3.79), we have

$$\begin{aligned} Gain(S, C_{S1}(k), Z(J_d, J_s)) = \frac{1}{mk+1}\Bigg[&\left\lfloor\frac{1+J_d}{2}\right\rfloor + n - l - 1 \\ &+ \sum_{I_d=1}^{k}\left(\left\lfloor\frac{I_d + J_d - 1}{2}\right\rfloor + n - l\right) \\ &+ \sum_{I_s=1, I_s\neq J_s}^{m}\sum_{I_d=1}^{k}\left(\left\lfloor\frac{-I_d + J_d + 1}{2}\right\rfloor + n - l - 1\right)\Bigg]. \end{aligned} \tag{3.80}$$

We see that the expected gain increases with J_d, so that the minimum for $J_d > k$ is $J_d = k+1$. Replacing J_d, expanding the summations and considering that $n = ml + 1$, the expected gain can be simplified to $Gain(S, C_{S1}(k), Z(k+1, J_s)) =$

$$\begin{cases} \frac{4k+2k^2-4l-2km+k^2m-4klm+4n+4kmn}{4(mk+1)}, & \text{if } k \text{ is even} \\ \frac{4k+2k^2-4l-2km+k^2m-4klm+4n+4kmn+m-2}{4(mk+1)}, & \text{if } k \text{ is odd} \end{cases} \quad (3.81)$$

(b) If $v_j = (J_d, J_s)$ with $J_d = 0$ and $J_s = 0$, replacing the expression of $a_{(I_d,I_s),(J_d,J_s)}$ from Corollary 3.14 in (3.79), we have

$$Gain(S, C_{S1}(k), Z(0,0)) = \frac{1}{mk+1}\left[m\sum_{I_d=1}^{k}\left(l+1-\left\lfloor\frac{I_d+2}{2}\right\rfloor\right)\right] \quad (3.82)$$

which can be simplified by expanding the summation to

$$Gain(S, C_{S1}(k), Z(0,0)) = \begin{cases} \frac{m}{mk+1}\left(kl - \frac{k^2}{4}\right), & \text{if } k \text{ is even} \\ \frac{m}{mk+1}\left(kl - \frac{k^2}{4} + \frac{1}{4}\right), & \text{if } k \text{ is odd.} \end{cases} \quad (3.83)$$

(c) If $v_j = (J_d, J_s)$ with $1 \leq J_d \leq k$ and $J_s \neq 0$;

(c.1) If $J_d = 1$, replacing the expression of $a_{(I_d,I_s),(J_d,J_s)}$ from Corollary 3.14 in (3.79), we have

$$\begin{aligned} Gain(S, C_{S1}(k), Z(1, J_s)) = \frac{1}{mk+1}&\left[n - l + \sum_{I_d=2}^{k}\left(l - \left\lfloor\frac{I_d+1}{2}\right\rfloor\right)\right. \\ &\left. +(m-1)\left(l + \sum_{I_d=2}^{k}\left(l + \left\lfloor\frac{-I_d+2}{2}\right\rfloor\right)\right)\right] \end{aligned} \quad (3.84)$$

which can be simplified to

$$Gain(S, C_{S1}(k), Z(1, J_s)) = \begin{cases} \frac{4n+4lkm+2km-k^2m-4k-8l+4}{4(1+km)}, & \text{if } k \text{ is even} \\ \frac{4n+4lkm+2km-k^2m-4k-8l+4+m}{4(1+km)}, & \text{if } k \text{ is odd.} \end{cases} \quad (3.85)$$

Considering the different cases, we can determine that this gain is greater than the one from (3.83).

(c.2) If $1 < J_d < k$, replacing the expression of $a_{(I_d,I_s),(J_d,J_s)}$ from Corollary 3.14 in (3.79), we have

$$\begin{aligned} Gain(S, C_{S1}(k), Z(J_d, J_s)) = \frac{1}{mk+1} \Bigg[\left\lfloor \frac{1+J_d}{2} \right\rfloor + n - l - 1 \\ + \sum_{I_d=1}^{J_d-1} \left(\left\lfloor \frac{I_d+J_d-1}{2} \right\rfloor + n - l \right) + \sum_{I_d=J_d+1}^{k} \left(l - \left\lfloor \frac{I_d+J_d}{2} \right\rfloor \right) \\ + \sum_{I_s=1, I_s \neq J_s}^{m} \left(\sum_{I_d=1}^{J_d-1} \left(\left\lfloor \frac{-I_d+J_d+1}{2} \right\rfloor + n - l - 1 \right) + l \right. \\ \left. + \sum_{I_d=J_d+1}^{k} \left(l + \left\lfloor \frac{-I_d+J_d+1}{2} \right\rfloor \right) \right) \Bigg] \end{aligned} \tag{3.86}$$

We want to show that this expected gain is greater than the one from case (b) where Player 2 chooses the body of the spider as starting vertex. Note that the vertices (I_d, I_s) of Player 1 for which $a_{(I_d,I_s),(J_d,J_s)} < a_{(I_d,I_s),(0,0)}$ are $\{(I_d, I_s) \mid J_d \leq I_d \leq k \text{ and } I_s = J_s\}$. Thus, by choosing the vertex (J_d, J_s) instead of $(0, 0)$, Player 2 only reduces the payoff to Player 1 on these vertices. Furthermore,

$$\begin{aligned} &\sum_{I_d=J_d}^{k} \left(a_{(I_s,I_d),(J_s,J_d)} - a_{(I_s,I_d),(0,0)} \right) \\ &= \begin{cases} \frac{J_d^2}{2} - \frac{J_d k}{2} - \frac{k}{4} + \frac{J_d}{4}, & \text{if } J_d \text{ is even and } k \text{ is even} \\ \frac{J_d^2}{2} - \frac{J_d k}{2} - \frac{k}{4} + \frac{J_d}{4} - \frac{1}{4}, & \text{if } J_d \text{ is even and } k \text{ is odd} \\ \frac{J_d^2}{2} - \frac{J_d k}{2} + \frac{k}{4} - \frac{J_d}{4} + \frac{1}{4}, & \text{if } J_d \text{ is odd and } k \text{ is even} \\ \frac{J_d^2}{2} - \frac{J_d k}{2} + \frac{k}{4} - \frac{J_d}{4}, & \text{if } J_d \text{ is odd and } k \text{ is odd.} \end{cases} \end{aligned} \tag{3.87}$$

On the other hand, we have the following:

$$a_{(0,0),(J_d,J_s)} - a_{(0,0),(0,0)} \geq n - l \geq (m-1)l, \tag{3.88}$$

for $I_s = 0$,

$$\sum_{I_d=1}^{l} \left(a_{(I_d,I_s),(J_d,J_s)} - a_{(I_d,I_s),(0,0)} \right) \geq 0 \tag{3.89}$$

for $I_s \neq J_s$, $I_s \neq 0$ and

$$\sum_{I_d=1}^{J_d-1} \left(a_{(I_d,I_s),(J_d,J_s)} - a_{(I_d,I_s),(0,0)} \right) \geq \sum_{I_d=1}^{J_d-1} (n - 2l) \geq (J_d - 1)(m-2)l \tag{3.90}$$

for $I_s = J_s,\ I_s \neq 0$.

Grouping these equations, we have

$$
\begin{aligned}
&(mk+1)\left[Gain(S, C_{S1}(k), Z(J_d, J_s)) - Gain(S, C_{S1}(k), Z(0,0))\right] \\
&= a_{(0,0),(J_d,J_s)} - a_{(0,0),(0,0)} + \sum_{I_s=1}^{m}\sum_{I_d=1}^{k}\left(a_{(I_d,I_s),(J_d,J_s)} - a_{(I_d,I_s),(0,0)}\right) \\
&\geq \begin{cases}
\frac{J_d^2}{2} - \frac{J_d k}{2} - \frac{k}{4} + \frac{J_d}{4} + (m-1)l + (J_d-1)(m-2)l, \\
\qquad\qquad \text{if } J_d \text{ is even and } k \text{ is even} \\
\frac{J_d^2}{2} - \frac{J_d k}{2} - \frac{k}{4} + \frac{J_d}{4} - \frac{1}{4} + (m-1)l + (J_d-1)(m-2)l, \\
\qquad\qquad \text{if } J_d \text{ is even and } k \text{ is odd} \\
\frac{J_d^2}{2} - \frac{J_d k}{2} + \frac{k}{4} - \frac{J_d}{4} + \frac{1}{4} + (m-1)l + (J_d-1)(m-2)l, \\
\qquad\qquad \text{if } J_d \text{ is odd and } k \text{ is even} \\
\frac{J_d^2}{2} - \frac{J_d k}{2} + \frac{k}{4} - \frac{J_d}{4} + (m-1)l + (J_d-1)(m-2)l, \\
\qquad\qquad \text{if } J_d \text{ is odd and } k \text{ is odd.}
\end{cases}
\end{aligned}
\tag{3.91}
$$

Considering separately the cases from (3.91), we can show that the difference is always positive. Thus, the expected gain of Player 1 when Player 2 chooses a vertex $v_j = (J_d, J_s)$ with $J_s \neq 0$ and $1 < J_d < k$ is greater than when Player 2 chooses the body of the spider.

(c.3) If $J_d = k$, replacing the expression of $a_{(I_d,I_s),(J_d,J_s)}$ from Corollary 3.14 in (3.79), we have

$$
\begin{aligned}
Gain(S, C_{S1}(k), Z(k, J_s)) = \frac{1}{mk+1}\Bigg[&\left\lfloor\frac{1+k}{2}\right\rfloor + n - l - 1 \\
&+ \sum_{I_d=1}^{k-1}\left(\left\lfloor\frac{I_d+k-1}{2}\right\rfloor + n - l\right) \\
&+(m-1)\left(\sum_{I_d=1}^{k-1}\left(\left\lfloor\frac{-I_d+k+1}{2}\right\rfloor + n - l - 1\right) + l\right)\Bigg].
\end{aligned}
\tag{3.92}
$$

Similarly as in case (c.2), we can show that this expected gain is greater than the one where Player 2 chooses the body of the spider. The payoff to Player 1 increases on all the vertices except $(I_d, I_s) = (k, J_s)$ for which the payoff decreases by $l - \left\lfloor \frac{k}{2} \right\rfloor$. On the other hand, the payoff on $(0, 0)$, increases by $\left\lfloor \frac{k}{2} \right\rfloor + n - l$ and

$$\left\lfloor \frac{k}{2} \right\rfloor + n - l > l - \left\lfloor \frac{k}{2} \right\rfloor \tag{3.93}$$

since $n - l > l$. Therefore, the expected gain is greater than the one from (3.83).

Finally, the expected gain when Player 2 chooses the body of the spider (see (3.83)) is less than the expected gain when Player 2 chooses a vertex (J_d, J_s) with $J_d = k + 1$ and $J_s \neq 0$ (see (3.81)). Therefore, the minimal gain that Player 1 can get with the strategy $C_{S1}(k)$ is when Player 2 chooses the body of the spider. Thus, the guaranteed gain of Player 1 with the strategy $C_{S1}(k)$ on S with m arms each having l vertices is

$$GGain(S, C_{S1}(k)) = \begin{cases} \frac{m}{mk+1}\left(kl - \frac{k^2}{4}\right), & \text{if } k \text{ is even} \\ \frac{m}{mk+1}\left(kl - \frac{k^2}{4} + \frac{1}{4}\right), & \text{if } k \text{ is odd.} \end{cases} \tag{3.94}$$

□

Now that we have an expression for the guaranteed gain of Player 1 with the strategy $C_{S1}(k)$, we want to maximize her guaranteed gain over k.

Lemma 3.19. *The optimal guaranteed gain of Player 1 with the safe strategy $C_{S1}(k)$ on S with m legs each having l vertices is $GGain(S, C_{S1}(k^*)) = l - \frac{\sqrt{l}}{\sqrt{m}} + \mathcal{O}(1)$ $(l \to \infty)$ where $k^* = \frac{2\sqrt{l}}{\sqrt{m}} + \mathcal{O}(1)$ $(l \to \infty)$ is the optimal integer k.*

Proof. From Lemma 3.18, we have the guaranteed gain of Player 1 with the strategy $C_{S1}(k)$ on S with m legs each having l vertices:

$$GGain(S, C_{S1}(k)) = \begin{cases} \frac{m}{mk+1}\left(kl - \frac{k^2}{4}\right), & \text{if } k \text{ is even} \\ \frac{m}{mk+1}\left(kl - \frac{k^2}{4} + \frac{1}{4}\right), & \text{if } k \text{ is odd.} \end{cases} \tag{3.95}$$

We want to maximize the guaranteed gain over k for m and l fixed. This is done in a similar manner as in the proof of Lemma 3.7, defining the functions

$$g_1(x) = \frac{m}{mx + 1}\left(xl - \frac{x^2}{4}\right) \tag{3.96}$$

and

$$g_2(x) = \frac{m}{mx + 1}\left(xl - \frac{x^2}{4} + \frac{1}{4}\right) \tag{3.97}$$

and considering that $g_2(x) = g_1(x) + \mathcal{O}(1)$ $(x \to \infty)$. □

Remark 3.20. When m, the number of legs of a spider with legs each having l vertices, tends to infinity, Player 1's guaranteed gain with the strategy $C_{S1}(k)$ with $k = 1$ approaches l.

Proof. From Lemma 3.18, we know that Player 1's guaranteed gain when she adopts the safe strategy $C_{S1}(k)$ on a spider with legs each having l vertices is

$$GGain(S, C_{S1}(k)) = \begin{cases} \frac{m}{mk+1}(kl - \frac{k^2}{4}), & \text{if } k \text{ is even} \\ \frac{m}{mk+1}(kl - \frac{k^2}{4} + \frac{1}{4}), & \text{if } k \text{ is odd} \end{cases} \tag{3.98}$$

If $k = 1$, we have

$$GGain(S, C_{S1}(k = 1)) = \left(\frac{m}{m+1}\right) l = \left(1 - \frac{1}{m+1}\right) l \approx l \tag{3.99}$$

when m tends to infinity. □

We now need a lemma on the maximal gain of Player 2 when Player 2 adopts the strategy $C_{S1}(k)$ on S.

Lemma 3.21. *The optimal maximal gain of Player 1 when Player 2 uses the opposing strategy $C_{S1}(k)$ on S with m legs each having l vertices is $MGain(S, C_{S1}(k^*)) = l$ where $k^* = 0$ is the optimal integer k.*

Proof. It is clear that if Player 2 adopts the strategy $C_{S1}(k)$ with $k = 0$, i.e. Player 2 chooses to start with the body of the spider, then the maximum gain that Player 1 can get is l, the number of vertices in a leg.
If Player 2 adopts the strategy $C_{S1}(k)$ with $k \neq 0$, then the expected gain of Player 1 when she chooses the body of the spider is

$$\begin{aligned} Gain(S, Z(0,0), C_{S1}(k)) &\geq \left(\frac{1}{mk+1}\right) 0 + \frac{mk}{mk+1}\left((m-1)l + 1\right) \\ &= \frac{mk(lm - l + 1)}{mk+1} \\ &> l \end{aligned} \tag{3.100}$$

for $m \geq 3$, $l \geq 1$ and $k \geq 1$. Thus, the optimal strategy $C_{S1}(k)$ for Player 2 is $k = 0$. □

We now have the necessary results to prove Theorem 3.17.

Proof. **(Theorem 3.17)** This proof is essentially the same as for Theorem 3.5 having the guaranteed gain from Lemma 3.19 and the maximal gain from Lemma 3.21. □

Let us compare the bounds on the safety value with the payoffs in the pure Nash equilibrium for spider with legs of equal length from Corollary 3.12.

Theorem 3.22. *For a spider S with $m \geq 3$ arms each having l vertices, the safety value of Player 1 asymptotically approaches the worst of the two gains in the two-player pure Nash equilibrium as the number of vertices in the legs, l, tends to infinity.*

Proof. This result directly follows from evaluating the limits of the bounds in Theorem 3.17 when l tends to infinity. □

Again, we have that the asymptotic safety value is related to the payoffs in the pure Nash equilibrium.

3.2.4 Bounds on Safety Value with the Strategy $C_{S2}(k)$

With the purpose of tightening the bounds on the safety value of Player 1 from Theorem 3.17 of the last section, we suggest the following strategy $C_{S2}(k)$. It is a modified version of the strategy $C_{S1}(k)$ of Definition 3.16. In the new strategy, the probability of choosing the body of the spider is distinct from the probability of choosing the first k vertices on the legs.

Definition 3.23. Let the strategy $C_{S2}(k)$ be a mixed strategy on a spider, S, with m legs each having l vertices where $C_{S2}(k) = (z_1, z_2, ..., z_n)$ and z_i, the probability of choosing the vertex $v_i = (I_d, I_s)$, with $I_d \in \{1, 2, ...l\}$ and $I_s \in \{s_1, s_2, ...,\}$ or $I_d = 0$ and $I_s = 0$, is

$$z_i = \begin{cases} 0, & \text{if } k < I_d \leq l \\ \beta = \frac{(m-1)l+1}{km^2l+km-lkm+l-k}, & \text{if } 0 < I_d \leq k \\ \alpha = \frac{l-k}{km^2l+km-lkm+l-k}, & \text{if } I_d = 0 \end{cases} \tag{3.101}$$

and $k \in \{1, 2, ..., l\}$.

As an example, Figure 3.11 illustrates the strategy $C_{S2}(k)$ with $k = 2$. That is, the strategy where the body is chosen with probability α, the first two vertices on each leg are chosen with probability β and the other vertices are chosen with probability

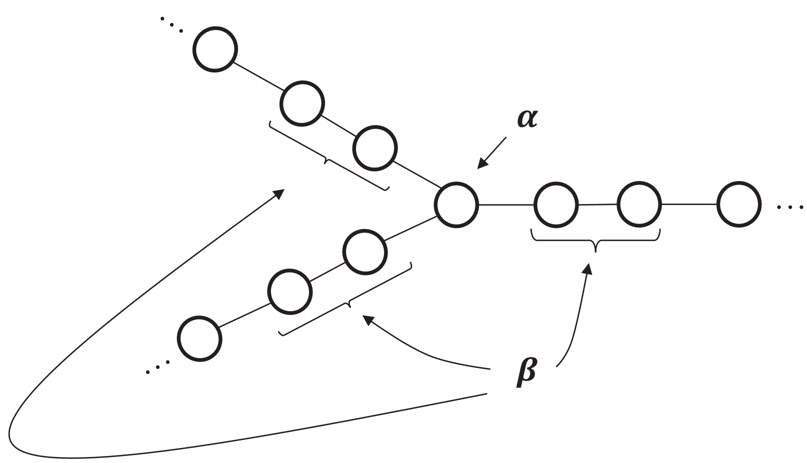


Figure 3.11: Illustration of the strategy $C_{S2}(k)$ with $k = 2$ on a spider with 3 legs.

0, where α and β are as defined in (3.101). We can consider the strategy $C_{S2}(k)$ as a safe strategy for Player 1 on a spider with legs of equal length to improve the lower bound on the safety value of Player 1. This leads to the following:

Theorem 3.24. *In the two-player game of Competitive Diffusion on a spider, S, with m legs each having l vertices, the safety value of Player 1 is between $l - \frac{\sqrt{l}}{\sqrt{m(m-1)}} + \mathcal{O}(1)$ $(l \to \infty)$ and l.*

In order to prove this theorem, we will consider some lemmas on the guaranteed gain of Player 1 with the safe strategy $C_{S2}(k)$. Before proceeding, let us first check that this new bound, $l - \frac{\sqrt{l}}{\sqrt{m(m-1)}} + \mathcal{O}(1)$ $(l \to \infty)$, is tighter than the one from Theorem 3.17, $l - \frac{\sqrt{l}}{\sqrt{m}} + \mathcal{O}(1)$ $(l \to \infty)$. The bound is tighter if

$$l - \frac{\sqrt{l}}{\sqrt{m(m-1)}} \geq l - \frac{\sqrt{l}}{\sqrt{m}} \tag{3.102}$$

and this is verified for $m \geq 3$ and $l \geq 1$.

Lemma 3.25. *Considering the strategy $C_{S2}(k)$ as a safe strategy for Player 1 on a spider, S, with m legs each having l vertices, we get*

$$GGain(S, C_{S2}) = \begin{cases} m\beta\left(kl - \frac{k^2}{4}\right), & \textit{if } k \textit{ is even} \\ m\beta\left(kl - \frac{k^2}{4} + \frac{1}{4}\right), & \textit{if } k \textit{ is odd} \end{cases} \tag{3.103}$$

where β is the probability that Player 1 chooses one of the first k vertices in one of the legs as in Definition 3.23.

Proof. Similarly as in the proof of Lemma 3.18, we have that the guaranteed gain of Player 1 is

$$GGain(S, C_{S2}(k)) = \min_j Gain(S, C_{S2}(k), Z(v_j)). \tag{3.104}$$

Furthermore, the expected gain of Player 1 when Player 2 chooses a vertex $v_j = (J_d, J_s)$ is

$$Gain(S, C_{S2}(k), Z(J_d, J_s)) = \alpha \cdot a_{(0,0),(J_d,J_s)} + \beta \sum_{I_s=1}^{m} \sum_{I_d=1}^{k} a_{(I_d,I_s),(J_d,J_s)} \tag{3.105}$$

where a_{ij} is the entry of the game matrix of Player 1 from Corollary 3.14 and

$$\alpha = \frac{l-k}{km^2l + km - lkm + l - k}, \quad \beta = \frac{(m-1)l+1}{km^2l + km - lkm + l - k} \tag{3.106}$$

are respectively the probability of choosing the body of the spider and the probability of choosing one of the first k vertices on the legs as defined in Definition 3.23. We have three cases to consider:

(a) If $v_j = (J_d, J_s)$ with $J_d > k$ and $J_s \neq 0$, then

$$Gain(S, C_{S2}(k), Z(J_d, J_s)) \geq n - l > l \tag{3.107}$$

since $a_{(I_d,I_s),(J_d,J_s)} \geq n - l$ for all $I_d \in \{0, 1, ..k\}$.

(b) If $J_d = 0$ and $J_s = 0$, i.e. Player 2 chooses the body of the spider, replacing the expression of $a_{(I_d,I_s),(J_d,J_s)}$ from Corollary 3.14 in (3.105), we have:

$$\begin{aligned} Gain(S, C_{S2}(k), Z(0,0)) &= m\beta \sum_{I_d=1}^{k} \left(l + 1 - \left\lfloor \frac{I_d+1}{2} \right\rfloor \right) \\ &= \begin{cases} m\beta \left(kl - \frac{k^2}{4} \right), & \text{if } k \text{ is even} \\ m\beta \left(kl - \frac{k^2}{4} + \frac{1}{4} \right), & \text{if } k \text{ is odd} \end{cases} \end{aligned} \tag{3.108}$$

by (3.83).

(c) If $1 \leq J_d \leq k$ and $J_s \neq 0$;

(c.1) If $J_d = 1$, replacing the expression of $a_{(I_d,I_s),(J_d,J_s)}$ from Corollary 3.14 in (3.105) gives

$$\begin{aligned} Gain(S, C_{S2}, Z(1, J_s)) = \alpha(n-l) + \beta \sum_{I_d=2}^{k} \left(l - \left\lfloor \frac{I_d+1}{2} \right\rfloor \right) \\ + (m-1)\beta \left(l + \sum_{I_d=2}^{k} \left(l + \left\lfloor \frac{-I_d+2}{2} \right\rfloor \right) \right) \end{aligned} \tag{3.109}$$

which can be simplified to

$$Gain(S, C_{S2}, Z(1, J_s)) = \begin{cases} \alpha(n-l) + (m-1)\beta l + m\beta \left(kl - \frac{k^2}{4} - l + \frac{k}{2} \right), \\ \qquad \text{if } k \text{ is even} \\ \alpha(n-l) + (m-1)\beta l + m\beta \left(kl - \frac{k^2}{4} - l + \frac{k}{2} - \frac{1}{4} \right), \\ \qquad \text{if } k \text{ is odd.} \end{cases} \tag{3.110}$$

Considering the different cases, we can determine that this gain is greater than the one from (3.108) with $k > 1, m \geq 3$.

(c.2) If $1 < J_d < k$, we have that

$$\begin{aligned} & Gain(S, C_{S2}(k), Z(J_d, J_s)) - Gain(S, C_{S2}(k), Z((0,0))) \\ \geq & \begin{cases} \alpha(m-1)l + \beta \left(\frac{J_d^2}{2} - \frac{J_d k}{2} - \frac{k}{4} + \frac{J_d}{4} + (J_d-1)(m-2)l \right), \\ \qquad \text{if } J_d \text{ is even and } k \text{ is even} \\ \alpha(m-1)l + \beta \left(\frac{J_d^2}{2} - \frac{J_d k}{2} - \frac{k}{4} + \frac{J_d}{4} - \frac{1}{4} + (J_d-1)(m-2)l \right), \\ \qquad \text{if } J_d \text{ is even and } k \text{ is odd} \\ \alpha(m-1)l + \beta \left(\frac{J_d^2}{2} - \frac{J_d k}{2} + \frac{k}{4} - \frac{J_d}{4} + \frac{1}{4} + (J_d-1)(m-2)l \right), \\ \qquad \text{if } J_d \text{ is odd and } k \text{ is even} \\ \alpha(m-1)l + \beta \left(\frac{J_d^2}{2} - \frac{J_d k}{2} + \frac{k}{4} - \frac{J_d}{4} + (J_d-1)(m-2)l \right), \\ \qquad \text{if } J_d \text{ is odd and } k \text{ is odd.} \end{cases} \end{aligned} \tag{3.111}$$

using the inequalities (3.87), (3.88), (3.89) and (3.90) from the proof of Lemma 3.18. Considering the cases separately, we can show that the difference is always positive. Thus, the expected gain of Player 1 when Player 2 chooses a vertex $v_j = (J_d, J_s)$ with $1 < J_d < k$ and $J_s \neq 0$ is greater than when Player 2 chooses the body of the spider.

(c.3) If $J_d = k$, we can show in a similar manner as case (c.2) that the expected gain is greater than the one when Player 2 chooses the body of the spider. The payoff to Player 1 increases on all the vertices except when $(I_d, I_s) = (k, J_s)$ for which the payoff decreases by $l - \left\lfloor \frac{k}{2} \right\rfloor \leq l$. The payoffs for all other vertices on which Player 1 has a positive probability were less than or equal to l when Player 2 chose the body of the spider and are now at least $n - l$. Thus, these payoffs increase by at least $n - 2l > (m-2)l$ and so

$$\begin{aligned} &Gain(S, C_{S2}(k), Z(k, J_s)) - Gain(S, C_{S2}(k), Z(0,0)) \\ &\qquad \geq (1-\beta)(m-2)l - \beta l \geq 0 \end{aligned} \tag{3.112}$$

since $m \geq 3$.

Finally, after considering all the cases, we see that the expected gain when Player 2 chooses the body of the spider (see (3.108)) is the minimal expected gain of Player 1. Therefore,

$$GGain(S, C_{S2}) = \begin{cases} m\beta\left(kl - \frac{k^2}{4}\right), & \text{if } k \text{ is even} \\ m\beta\left(kl - \frac{k^2}{4} + \frac{1}{4}\right), & \text{if } k \text{ is odd} \end{cases} \tag{3.113}$$

□

Now that we have an expression for the guaranteed gain of Player 1 with the strategy $C_{S2}(k)$, we want to maximize her guaranteed gain over k.

Lemma 3.26. *The optimal guaranteed gain of Player 1 with the strategy $C_{S2}(k)$ on a spider, S, with m legs each having l vertices is*

$$GGain(S, C_{S2}(k^*)) = l - \frac{\sqrt{l}}{\sqrt{m(m-1)}} + \mathcal{O}(1) \ (l \to \infty) \tag{3.114}$$

where $k^* = \frac{2\sqrt{l}}{\sqrt{m(m-1)}} + \mathcal{O}(1)$ $(l \to \infty)$ is the optimal integer k.

Proof. From Lemma 3.25, we have the guaranteed gain of Player 1 with the strategy $C_{S2}(k)$ on S with m legs each having l vertices:

$$GGain(S, C_{S2}(k)) = \begin{cases} m\beta\left(kl - \frac{k^2}{4}\right), & \text{if } k \text{ is even} \\ m\beta\left(kl - \frac{k^2}{4} + \frac{1}{4}\right), & \text{if } k \text{ is odd} \end{cases} \tag{3.115}$$

where β is the probability as defined in Definition 3.23. Replacing β, we have

$$GGain(S, C_{S2}(k)) = \begin{cases} m\left(\frac{(m-1)l+1}{km^2l+km-lkm+l-k}\right)\left(kl - \frac{k^2}{4}\right), & \text{if } k \text{ is even} \\ m\left(\frac{(m-1)l+1}{km^2l+km-lkm+l-k}\right)\left(kl - \frac{k^2}{4} + \frac{1}{4}\right), & \text{if } k \text{ is odd} \end{cases} \tag{3.116}$$

We want to maximize the guaranteed gain over k for m and l fixed. This is done in a similar manner as in the proof of Lemma 3.7, defining the functions

$$g_1(x) = m\left(\frac{(m-1)l+1}{xm^2l + xm - lxm + l - x}\right)\left(xl - \frac{x^2}{4}\right) \tag{3.117}$$

and

$$g_2(x) = m\left(\frac{(m-1)l+1}{xm^2l + xm - lxm + l - x}\right)\left(xl - \frac{x^2}{4} + \frac{1}{4}\right), \tag{3.118}$$

considering that $g_2(x) = g_1(x) + \mathcal{O}(1)$ $(x \to \infty)$ and that $g_1(x)$ reaches a maximum at $x^* = \frac{2\sqrt{l}}{\sqrt{m(m-1)}} + \mathcal{O}(1)$ $(l \to \infty)$. □

We can now prove Theorem 3.24.

Proof. **(Theorem 3.24)** For the lower bound, the proof is the same as Theorem 3.5, having the guaranteed gain from Lemma 3.26. The upper bound l is simply taken from Theorem 3.17. □

As a last point, let us explain quickly how the probabilities of α and β were suggested since similar ideas will be used to suggest mixed strategies for the players in the next chapters. Consider a mixed strategy X for Player 1 on S where the probability of choosing the body of the spider is α, the probability of choosing each of the first k vertices on the legs of S is β and the probability of choosing the other vertices is zero. If Player 2 chooses the body of the spider, then

$$Gain(S, X, Z(0,0)) \geq \alpha \cdot 0 + (1-\alpha)(l-k) \tag{3.119}$$

since $a_{(I_d,I_s),(0,0)} = l - \left\lfloor \frac{I_d}{2} \right\rfloor \geq (l-k)$ for $1 \leq I_d \leq k$. On the other hand, if Player 2 chooses a vertex (J_d, J_s) with $1 \leq J_d \leq k$,

$$Gain(S, X, Z(J_d, J_s)) \geq \alpha(n-l) + (1-\alpha-\beta)(l-k) \tag{3.120}$$

since $a_{(0,0),(J_d,J_s)} = \left\lfloor \frac{1+J_d}{2} \right\rfloor + n - l - 1 \geq (n-l)$ and $a_{(I_d,I_s),(J_d,J_s)} \geq (l-k)$ for the other cases (see Corollary 3.14).

Since Player 1 does not want to give Player 2 an advantage of choosing the body of the spider over another vertex on which she assigns a positive probability, it makes sense to equate these two expected gains and solve for α and β using $\alpha + km\beta = 1$. The solution of α and β is precisely the expressions in Definition 3.23. This idea of considering the expected gain in two scenarios and determining the probabilities that makes them equal, will be used to suggest strategies to the players for the game of Competitive Diffusion on Complete Trees in the next chapter.

Chapter 4

Complete Trees

In this chapter, we will study the two-player safe game of Competitive Diffusion on complete trees. Recall from Definition 1.39 that a complete m-ary tree of height h, $T(m, h)$ is a rooted tree in which all the internal vertices have m children and all the leaves have depth h. Therefore, the number of vertices in $T(m, h)$ is $n = \frac{m^{h+1}-1}{m-1}$.

4.1 Pure Nash Equilibrium

Since we have the two-player pure Nash equilibrium for trees in Theorem 2.5 in terms of the centroid, we only need to determine what is the centroid of a complete m-ary tree.

Proposition 4.1. *The centroid of a complete m-ary tree of height h, which has $n = \frac{m^{h+1}-1}{m-1}$ vertices, is the root of the tree and the weight of the centroid is $\frac{m^h-1}{m-1}$.*

Proof. Let r be the root of $T(m, h)$ and consider the branches at r (see Definition 1.48). There are m of them and they each have $\frac{n-1}{m}$ edges. Since $\frac{n-1}{m} \leq \frac{n}{2}$ for $m \geq 2$, we have from Theorem 1.51, that r is a centroid vertex. Moreover, we know from Theorem 1.50 that if two centroid vertices exist, they are adjacent. Thus, the only possible other centroid vertex would be a vertex in level 1. However, the vertices in level 1 have a branch with $(m-1)\left(\frac{n-1}{m}\right) + 1 \geq \frac{n}{2}$ edges and thus are not centroid vertices by Theorem 1.51. Therefore, the centroid of a complete m-ary tree of height h is the root and its weight is $\frac{n-1}{m} = \frac{m^h-1}{m-1}$ since $n = \frac{m^{h+1}-1}{m-1}$. □

Corollary 4.2. *In the two-player game of Competitive Diffusion on a complete m-ary tree of height h, a pure Nash equilibrium is attained when one player has the root of the tree as starting vertex and the other player has one of the vertices in level 1. Moreover, the payoffs to the players are m^h and $\frac{m^h-1}{m-1}$ respectively.*

Proof. This result immediately follows from Theorem 2.5 having proved in Proposition 4.1 that the root is the centroid. □

4.2 Safety Value of Player 1 on Complete m-ary trees

Let us consider strategies which have positive probabilities on the root and the vertices in the first level of $T(m, h)$. Following a similar idea as with the strategy $C_{S2}(k)$ on spiders, we can assume that Player 1 chooses the root of the tree with probability α and the vertices in level 1 each with probability β. We can then determine Player 1's expected gains when Player 2 chooses the root and when Player 2 chooses a vertex in level 1. Then, we can force the two expected gains to be equal by solving for α and β knowing that $\alpha + m\beta = 1$. Similarly, we can assume that Player 2 chooses the root of the tree with probability α and the vertices in level 1 each with probability β, determining Player 1's expected gains when she chooses the root and when she chooses a vertex in level 1. Again, we can force the two expected gains to be equal by solving for α and β knowing that $\alpha + m\beta = 1$. This leads to the following two suggested strategies.

Let us identify a vertex v_i chosen by Player 1 as an ordered pair (I_d, I_e) where I_d is the distance of the vertex to the root or equivalently the level of the vertex and I_e is the position of the vertex in level I_d if the vertices are numbered from left to right by $\{0, 1, 2, ..., m^{I_d} - 1\}$ (see Figure 4.1). By convention, the root of the tree will be identified by the ordered pair $(0, 0)$. Similarly, a vertex v_j of Player 2 will be denoted by (J_d, J_e).

Note that with this notation, one could determine if a vertex $v_i = (I_d, I_e)$ is a descendant of a vertex $v_j = (J_d, J_e)$ by checking if

$$I_e \in \{J_e m^{I_d - J_d}, J_e m^{I_d - J_d} + m^{I_d - J_d} - 1\} \tag{4.1}$$

or not. Furthermore, recall from Definition 2.3, that the mixed strategy for which a player chooses a vertex v_i with probability 1 is denoted by $Z(v_i)$. For complete trees, it can equivalently be denoted by $Z((I_d, I_e))$ if the vertex v_i is identified by the ordered pair (I_d, I_e). In order to simplify the notation, we will write $Z(I_d, I_e)$ instead of $Z((I_d, I_e))$ as it should not create any confusion.

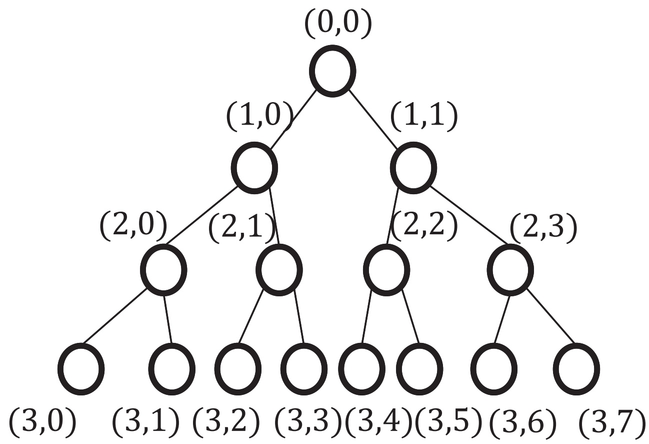


Figure 4.1: Labelling of the vertices in a complete tree.

Definition 4.3. Let the **strategy** μ_1 be a mixed strategy on $T(m,h)$ where $\mu_1 = (x_1, x_2, ..., x_n)$ and x_i, the probability of choosing the vertex $v_i = (I_d, I_e)$, is

$$x_i = \begin{cases} \alpha = \frac{m^h - 1}{m^{h+2} - m^{h+1} + m^h - 1}, & \text{if } I_d = 0 \\ \beta = \frac{(m-1)m^h}{m^{h+2} - m^{h+1} + m^h - 1}, & \text{if } I_d = 1 \\ 0, & \text{if } 2 \leq I_d \leq h \end{cases} \tag{4.2}$$

for $1 \leq i \leq n$.

Definition 4.4. Let the **strategy** μ_2 be a mixed strategy on $T(m,h)$ where $\mu_2 = (y_1, y_2, ..., y_n)$ and y_j, the probability of choosing a vertex $v_j = (J_d, J_e)$, is

$$y_j = \begin{cases} \alpha = \frac{(m-1)(m^{h+1} - m^h + 1)}{m^{h+2} - m^{h+1} + m^h - 1}, & \text{if } J_d = 0 \\ \beta = \frac{m^h - 1}{m^{h+2} - m^{h+1} + m^h - 1}, & \text{if } J_d = 1 \\ 0, & \text{if } 2 \leq J_d \leq h \end{cases} \tag{4.3}$$

for $1 \leq j \leq n$.

We consider the strategy μ_1 as a safe strategy for Player 1 on $T(m,h)$ and the strategy μ_2 as an opposing strategy for Player 2 on $T(m,h)$. This leads to the following results:

Theorem 4.5. *In the two-player game of Competitive Diffusion on the complete m-ary tree of height h, $T(m,h)$, the safety value of Player 1 is*

$$\frac{(n-1)((m-1)n+1)}{n(m^2-m+1)+m-1} \tag{4.4}$$

where $n = \frac{m^{h+1}-1}{m-1}$.

In order to prove this theorem, we will consider some lemmas on the guaranteed gain of Player 1 with the safe strategy μ_1 and the maximal gain of Player 1 when Player 2 has the opposing strategy μ_2.

Lemma 4.6. *The guaranteed gain of Player 1 with the strategy μ_1 on a complete m-ary tree of height h is*

$$GGain(T(m,h),\mu_1) = \frac{m^{h+1}(m^h-1)}{m^{h+2}-m^{h+1}+m^h-1}. \tag{4.5}$$

Proof. From Definition 2.9, we have that the guaranteed gain of Player 1 with the mixed strategy μ_1 on $T(m,h)$ is

$$GGain(T(m,h),\mu_1) = \min_j Gain(T(m,h),\mu_1,Z(v_j)) \tag{4.6}$$

where $1 \leq j \leq n$. Due to symmetry in $T(m,h)$ and in the strategy μ_1, we only need to consider the root of the tree and one vertex of each level as possible starting vertices for Player 2.

(a) If $v_j = (0,0)$, i.e. Player 2 chooses the root of the tree, the expected gain of Player 1 is

$$\begin{aligned} Gain(T(m,h),\mu_1,Z(0,0)) = {} & \alpha \cdot Gain\left(T(m,h),Z(0,0),Z(0,0)\right) \\ & + \beta \sum_{I_e=0}^{m-1} Gain(T(m,h),Z(1,I_e),Z(0,0)). \end{aligned} \tag{4.7}$$

Since

$$Gain\left(T(m,h),Z(0,0),Z(0,0)\right) = 0 \tag{4.8}$$

and

$$Gain(T(m,h),Z(1,I_e),Z(0,0)) = \sum_{k=0}^{h-1} m^k = \frac{m^h-1}{m-1}, \tag{4.9}$$

for $0 \leq I_e \leq m-1$, we have

$$Gain(T(m,h), \mu_1, Z(0,0)) = \alpha \cdot 0 + m\beta \left(\frac{m^h - 1}{m-1} \right). \tag{4.10}$$

Replacing the values of α and β from Definition 4.3 gives

$$Gain(T(m,h), \mu_1, Z(0,0)) = \frac{m^{h+1}(m^h - 1)}{m^{h+2} - m^{h+1} + m^h - 1}. \tag{4.11}$$

(b) If $v_j = (J_d, J_e)$ with $J_d = 1$, $0 \leq J_e \leq m-1$,

$$\begin{aligned} Gain(T(m,h), \mu_1, Z(1,J_e)) = {} & \alpha \cdot Gain\left(T(m,h), Z(0,0), Z(1,J_e)\right) \\ & + \beta \sum_{I_e=0}^{m-1} Gain(T(m,h), Z(1,I_e), Z(1,J_e)). \end{aligned} \tag{4.12}$$

Since

$$Gain\left(T(m,h), Z(0,0), Z(1,J_e)\right) = (m-1)\left(\frac{m^h - 1}{m-1} \right) + 1 = m^h \tag{4.13}$$

and

$$Gain(T(m,h), Z(1,I_e), Z(1,J_e)) = \frac{m^h - 1}{m-1}, \tag{4.14}$$

for $0 \leq I_e \leq m-1$, $I_e \neq J_e$ we have

$$Gain(T(m,h), \mu_1, Z(1,J_e)) = \alpha m^h + (m-1)\beta \left(\frac{m^h - 1}{m-1} \right). \tag{4.15}$$

Replacing the values of α and β from Definition 4.3 gives

$$Gain(T(m,h), \mu_1, Z(J_d,J_e)) = \frac{m^{h+1}(m^h - 1)}{m^{h+2} - m^{h+1} + m^h - 1}. \tag{4.16}$$

(c) If $v_j = (J_d, J_e)$ with $2 \leq J_d \leq h$,

$$\begin{aligned} Gain(T(m,h), \mu_1, Z(J_d,J_e)) = {} & \alpha \cdot Gain\left(T(m,h), Z(0,0), Z(J_d,J_e)\right) \\ & + \beta \sum_{I_e=0}^{m-1} Gain(T(m,h), Z(1,I_e), Z(J_d,J_e)). \end{aligned} \tag{4.17}$$

$Gain\left(T(m,h), Z(0,0), Z(J_d,J_e)\right) \geq m^h$ and $Gain(T(m,h), Z(1,I_e), Z(J_d,J_e)) \geq m^h$ since Player 2's starting vertex is further from the root than $(0,0)$ and

$(1, I_e)$ for $0 \le I_e \le m-1$. Therefore, the payoff to Player 1 will be at least the number of vertices in $m-1$ of the m branches. Thus,

$$Gain(T(m,h), \mu_1, Z(J_d, J_e)) \ge \alpha m^h + \beta \sum_{I_e=0}^{m-1} m^h = m^h. \tag{4.18}$$

Since $m^h > \frac{m^{h+1}(m^h-1)}{m^{h+2}-m^{h+1}+m^h-1}$, the guaranteed gain of Player 1 is

$$GGain(T(m,h), \mu_1) = \frac{m^{h+1}(m^h-1)}{m^{h+2}-m^{h+1}+m^h-1}. \tag{4.19}$$

□

Lemma 4.7. *The maximal gain of Player 1 when Player 2 uses the opposing strategy μ_2 on an m-ary tree of height h is*

$$MGain(T(m,h), \mu_2) = \frac{m^{h+1}(m^h-1)}{m^{h+2}-m^{h+1}+m^h-1} \tag{4.20}$$

Proof. Recall from Definition 2.10 that the maximal gain of Player 1 against the strategy μ_2 for Player 2 is

$$MGain(T(m,h), \mu_2) = \max_i Gain(T(m,h), Z(v_i), \mu_2) \tag{4.21}$$

where $1 \le i \le n$. In other words, it is the maximal expected gain Player 1 can get over all her possible starting vertices. Again, because of symmetry, we only need to consider the root of the tree and one vertex of each level as possible starting vertices for Player 1.

(a) If $v_i = (0,0)$, i.e. Player 1 chooses the root of the tree as starting vertex, her expected gain is

$$\begin{aligned} Gain(T(m,h), Z(0,0), \mu_2) = {} & \alpha \cdot Gain(T(m,h), Z(0,0), Z(0,0)) \\ & + \beta \sum_{J_e=0}^{m-1} Gain(T(m,h), Z(0,0), Z(1, J_e)). \end{aligned} \tag{4.22}$$

Since $Gain(T(m,h), Z(0,0), Z(0,0)) = 0$ and $Gain(T(m,h), Z(0,0), Z(1, J_e)) = (m-1)\sum_{k=0}^{h-1} m^k = m^h$, for $0 \le J_e \le m-1$, we have

$$Gain(T(m,h), Z(0,0), \mu_2) = \alpha \cdot 0 + m\beta m^h. \tag{4.23}$$

Replacing the values of α and β from Definition 4.4 gives

$$Gain(T(m,h), Z(0,0), \mu_2) = \frac{m^{h+1}(m^h-1)}{m^{h+2}-m^{h+1}+m^h-1} \tag{4.24}$$

(b) If $v_i = (I_d, I_e)$ with $I_d = 1$, $0 \leq I_e \leq m-1$,

$$\begin{aligned} Gain(T(m,h), Z(1,I_e), \mu_2) &= \alpha \cdot Gain(T(m,h), Z(1,I_e), Z(0,0)) \\ &\quad + \beta \sum_{J_e=0}^{m-1} Gain(T(m,h), Z(1,I_e), Z(1,J_e)). \end{aligned} \tag{4.25}$$

Since

$$Gain(T(m,h), Z(1,I_e), Z(0,0)) = \frac{m^h - 1}{m-1} \tag{4.26}$$

and

$$Gain(T(m,h), Z(1,I_e), Z(1,J_e)) = \frac{m^h - 1}{m-1} \tag{4.27}$$

for $0 \leq J_e \leq m-1$, $J_e \neq I_e$, we have

$$Gain(T(m,h), Z(1,I_e), \mu_2) = \alpha \left(\frac{m^h - 1}{m-1}\right) + (m-1)\beta \left(\frac{m^h - 1}{m-1}\right). \tag{4.28}$$

Replacing the values of α and β from Definition 4.4 gives

$$Gain(T(m,h), Z(1,I_e), \mu_2) = \frac{m^{h+1}(m^h - 1)}{m^{h+2} - m^{h+1} + m^h - 1}. \tag{4.29}$$

(c) If $v_i = (I_d, I_e)$ with $2 \leq I_d \leq h$,

$$\begin{aligned} Gain(T(m,h), Z(I_d,I_e), \mu_2) &= \alpha \cdot Gain(T(m,h), Z(I_d,I_e), Z(0,0)) \\ &\quad + \beta \sum_{J_e=0}^{m-1} Gain(T(m,h), Z(I_d,I_e), Z(1,J_e)). \end{aligned} \tag{4.30}$$

However, we have

$$Gain(T(m,h), Z(I_d,I_e), Z(0,0)) \leq \frac{m^{h-1} - 1}{m-1} \tag{4.31}$$

and

$$Gain(T(m,h), Z(I_d,I_e), Z(1,J_e)) \leq \frac{m^{h-1} - 1}{m-1}, \qquad \text{if } v_i \text{ is a descendant of } v_j \tag{4.32}$$

since in these cases, Player 1's payoff is less than or equal to the number of vertices in a subtree of the m-ary tree with the root at level 2. Moreover,

$$Gain(T(m,h), Z(I_d,I_e), Z(1,J_e)) \leq \frac{m^h - 1}{m-1}, \qquad \text{if } v_i \text{ is not a descendant of } v_j \tag{4.33}$$

since in this case, the payoff to Player 1 is less than or equal to the number of vertices in a subtree of the m-ary tree with the root at level 1. Thus,

$$Gain(T(m,h), Z(I_d, I_e), \mu_2) \leq (\alpha+\beta)\left(\frac{m^{h-1}-1}{m-1}\right) + (m-1)\beta\left(\frac{m^h-1}{m-1}\right). \tag{4.34}$$

Replacing the values of α and β from Definition 4.4 gives

$$\begin{aligned} Gain(T(m,h), Z(I_d, I_e), \mu_2) &\leq \frac{2m^{2h+1} - 3m^{2h} + 2m^{2h-1} + m^h - 2m^{h-1} - m^{h+2} + 1}{(m-1)(m^{h+2} - m^{h+1} + mh - 1)} \\ &< \frac{m^{h+1}(m^h-1)}{m^{h+2} - m^{h+1} + m^h - 1}. \end{aligned} \tag{4.35}$$

Therefore, the maximal gain of Player 1 against the strategy μ_2 is

$$MGain(T(m,h), \mu_2) = \frac{m^{h+1}(m^h-1)}{m^{h+2} - m^{h+1} + m^h - 1}. \tag{4.36}$$

□

We now prove Theorem 4.5.

Proof. **(Theorem 4.5)** This proof is essentially the same as the proof of Theorem 3.5 having the guaranteed gain from Lemma 4.6 and the maximal gain from Lemma 4.7 except that in this case, the bounds are equal. Thus, we have that the safety value of Player 1 is exactly $\frac{m^{h+1}(m^h-1)}{m^{h+2}-m^{h+1}+m^h-1}$. Since $n = \frac{m^{h+1}-1}{m-1}$, this is equivalent in terms of n and m to $\frac{(n-1)((m-1)n+1)}{n(m^2-m+1)+m-1}$. □

Let us compare the bounds on the safety value with the payoffs in the pure Nash equilibrium described in Corollary 4.2.

Theorem 4.8. *For a complete m-ary tree of height h with $n = \frac{m^{h+1}-1}{m-1}$ vertices, the safety value of Player 1 is asymptotically $\left(\frac{m-1}{m^2-m+1}\right)n + \mathcal{O}(1)$ $(n \to \infty)$ as the total number of vertices, n, tends to infinity and m is constant.*

Proof. This result immediately follows from evaluating the limit of the safety value of Theorem 4.5 as n tends to infinity and m is constant. □

In contrast to the path and the spider with legs of equal length, the safety value is not asymptotically equal to the payoffs in the pure Nash equilibrium which are $\frac{(m-1)n+1}{m} = \left(\frac{m-1}{m}\right)n + \mathcal{O}(1)$ $(n \to \infty)$ and $\frac{n-1}{m} = \left(\frac{1}{m}\right)n + \mathcal{O}(1)$ $(n \to \infty)$.

4.3 Special Case: Binary Tree

In the safe game of the two-player Competitive Diffusion on a complete binary tree, the safety value of Player 1 is asymptotically $\frac{n}{3}+\mathcal{O}(1)$ $(n \to \infty)$ whereas the payoffs in the pure Nash equilibrium are $\frac{n+1}{2}$ and $\frac{n-1}{2}$. Here, the payoffs in the pure Nash equilibrium are the same as the ones for paths with an odd number of vertices. Moreover, for paths, we had that the asymptotic safety value approaches $\frac{n}{2}+\mathcal{O}(1)$ $(n \to \infty)$. However this is not the case for the complete binary tree. The reason for this lies in the structure of the branches of the root. On the path, as the number of vertices increases, there are more and more vertices near the middle of the path on which the payoff is at least $\frac{n}{2}+\mathcal{O}(1)$ $(n \to \infty)$ no matter the chosen vertex of the opponent given that it is not the same one. Thus, Player 1 can use this to her advantage by choosing a strategy that has positive probabilities on these vertices. With a large number of them, the probability that Player 2 chooses the exact same one is small and therefore, the guaranteed gain of Player 1 approaches $\frac{n}{2}+\mathcal{O}(1)$ as n tends to infinity. On the other hand, increasing the number of vertices in the binary tree does not increase the number of vertices which are attractive to choose as starting vertex. Choosing a vertex in level 2 gives a payoff of only $\approx \frac{n}{4}$ when Player 2 chooses the root, no matter the height of the tree. Thus, the attractive vertices to choose from remain the root and the vertices of level 1. In this case, Player 1 cannot distribute the probabilities on more vertices to reduce the chance of being caught by Player 2 without causing a significant reduction of her expected payoff.

4.4 Special Case: Ternary Tree

In the safe game of the two-player Competitive Diffusion on a complete ternary tree, the safety value of Player 1 is asymptotically $\frac{2n}{7}+\mathcal{O}(1)$ $(n \to \infty)$ whereas the payoffs in the pure Nash equilibrium are $\frac{2n+1}{3}$ and $\frac{n-1}{3}$. Compared to the complete binary tree, the asymptotic safety value of Player 1 is closer to the worst of the two payoffs in the pure Nash equilibrium. This will also be the case with complete trees of higher degree as the difference

$$\left(\frac{n-1}{m}\right)-\left(\left(\frac{m-1}{m^2-m+1}\right)n+\mathcal{O}(1)\right)=\left(\frac{1}{m(m^2-m+1)}\right)n+\mathcal{O}(1)\ (n\to\infty) \tag{4.37}$$

approaches zero as m tends to infinity.

Chapter 5

Trees in General

In this chapter, we study the two-player safe game of Competitive Diffusion on trees in general. Recall from Definition 1.34 that a tree is any graph that contains no cycles. Hence, the paths, spiders and complete trees of Chapters 3 and 4 are specific cases of trees. This chapter presents a general approach to determine a safe strategy for Player 1 on any tree. Recall from Definition 1.48 and Theorem 1.50, that the centroid of a tree is the set of vertices with the minimal weight and it consists of either a single vertex (centroidal tree) or two adjacent vertices (bicentroidal tree). Moreover, given a tree with n vertices, the weights of the vertices and correspondingly the centroid of the tree can be determined by a linear time algorithm (see for example [13]). Therefore, in the following, we will assume that we know the weights of the vertices of any given tree as well as its centroid. Furthermore, we will consider two different classes for the trees, the centroidal trees and the bicentroidal trees.

5.1 Weights of the Vertices in a Tree

First, let us start with a few observations on the weights of the vertices in a tree.

Notation 5.1. Let v be a vertex in a tree with n vertices. Recall from Definition 1.48 that the weight of v, i.e. the number of edges in the largest branch at v, is denoted by $w(v)$. Moreover, we will denote $n - w(v)$ by $\overline{w}(v)$.

Lemma 5.2. *Let v be a vertex not part of the centroid of a tree T with n vertices. The weight of the vertex v is the number of edges in the branch at v in which lies the centroid of T.*

Proof. Recall from Definition 1.48 that the weight of the vertex v is the maximum number of edges in any branch at v. By way of contradiction, suppose that B, the branch at v in which lies the centroid, is not the branch with the maximum number of edges and let c be a vertex in the centroid.

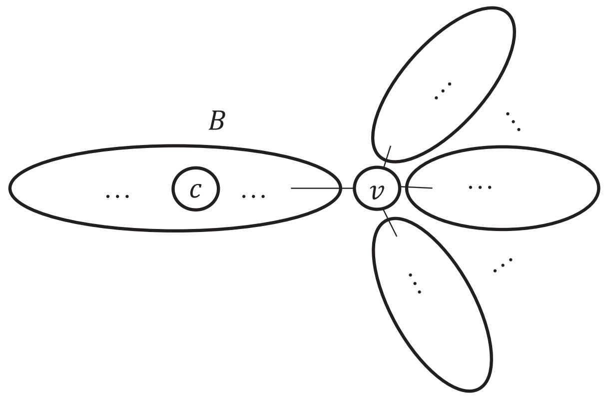


Figure 5.1: Illustration in the proof of Lemma 5.2.

Since the weight of a vertex is the maximum number of edges in one of its branches, we have

$$w(c) \geq n - |B|. \tag{5.1}$$

On the other hand, since B is not the branch at v with the maximum number of edges, we have

$$w(v) \leq n - |B|. \tag{5.2}$$

Thus,

$$n - |B| \leq w(c) \leq w(v) \leq n - |B| \tag{5.3}$$

since the centroid is the vertex with the minimal weight in T. Hence,

$$w(c) = w(v) \tag{5.4}$$

which is a contradiction since v is not a vertex in the centroid of T. Thus, the branch B is the branch at v with the maximum number of edges and hence the weight of v is precisely the number of edges in B. □

Lemma 5.3. *If v is a vertex not part of the centroid of a tree T with n vertices, then*

$$w(v) > \overline{w}(v) \tag{5.5}$$

where $\overline{w}(v) = n - w(v)$.

Proof. From Theorem 1.51, we know that a vertex is a centroid vertex if and only if all of its branches have less than or equal to $\frac{n}{2}$ vertices. Thus, if v is not a centroid vertex, at least one of its branches must have more than $\frac{n}{2}$ vertices. Since $w(v)$ is the number of edges in the largest branch at v, we must have

$$w(v) > \frac{n}{2}. \tag{5.6}$$

This is equivalent to

$$2w(v) > n \Leftrightarrow w(v) > n - w(v). \tag{5.7}$$

□

Lemma 5.4. *Consider a tree on n vertices T with centroid $C(T) = \{c\}$ if T is centroidal, or centroid $C(T) = \{c_1, c_2\}$ if T is bicentroidal. Let v_1, v_2 be two adjacent vertices not part of the centroid of T such there is a $v_2 - v_1 - c$ path in T if T is centroidal or a $v_2 - v_1 - c_i$ path in T if T is bicentroidal where $c_i \in \{c_1, c_2\}$ is the vertex in the centroid with the minimal distance to v_1. Let $N(v_1, v_2)$ be the total number of vertices on branches of v_1 other than the one in which lies the centroid and the one in which lies the vertex v_2. We have $N(v_1, v_2) = w(v_2) - w(v_1) - 1$.*

Proof. We know from Lemma 5.2 that the weights of v_1 and v_2 are the number of edges in the branches in which lies the centroid.

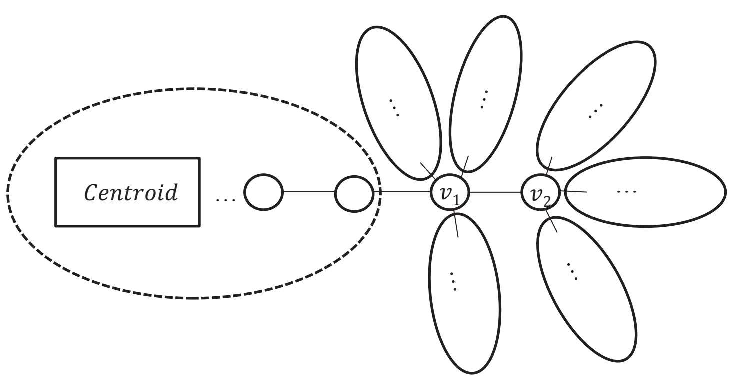


Figure 5.2: Illustration in the proof of Lemma 5.4.

Since the path from v_2 to the centroid is through the vertex v_1, the branch at v_2 in which lies the centroid includes the edges in the branch at v_1 in which lies the centroid. Therefore,

$$w(v_2) = w(v_1) + N(v_1, v_2) + 1. \tag{5.8}$$

where $N(v_1, v_2)$ is the number of edges in the branches at v_1 other than the one in which lies the centroid and the one in which lies the vertex v_2 and where the $+1$ corresponds to the edge $v_1 - v_2$. Rearranging, we have

$$N(v_1, v_2) = w(v_2) - w(v_1) - 1. \tag{5.9}$$

□

5.2 Centroidal Trees

Let us start by studying the Competitive Diffusion on centroidal trees. A centroidal tree with n vertices, ${}_{C}T(n)$, has one vertex as centroid and $n-1$ vertices distributed amongst branches at the centroid.

5.2.1 Types of Branches at the Centroid

We will distinguish three different types of branches at the centroid.

Definition 5.5. A **thick branch** at the centroid is a branch for which we have

$$w_2 \geq n - w_1 + \frac{w_1^2}{n} \tag{5.10}$$

where w_2 is the second lowest weight in the branch and w_1 is the lowest weight in the branch.

Definition 5.6. A **medium branch** at the centroid is a branch for which we have

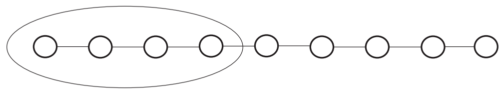
$$w_2 < n - w_1 + \frac{w_1^2}{n} \text{ and } w_3 \geq n - w_2 + \frac{w_2^2 + (w_2 - w_1)^2}{n + (w_2 - w_1)} \tag{5.11}$$

where w_3 is the third lowest weight in the branch, w_2 is the second lowest weight in the branch and w_1 is the lowest weight in the branch.

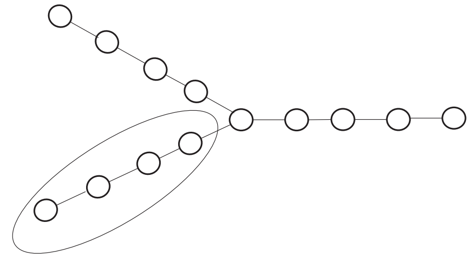
Definition 5.7. A **thin branch** at the centroid is a branch for which we have

$$w_2 < n - w_1 + \frac{w_1^2}{n} \text{ and } w_3 < n - w_2 + \frac{w_2^2 + (w_2 - w_1)^2}{n + (w_2 - w_1)} \tag{5.12}$$

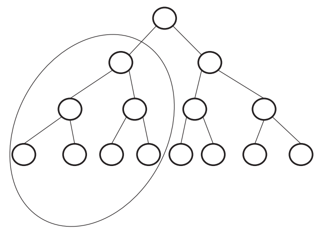
where w_3 is the third lowest weight in the branch, w_2 is the second lowest weight in the branch and w_1 is the lowest weight in the branch.



(a) Path.



(b) Spider.



(c) Complete m-ary tree.

Figure 5.3: Example of branches in graphs.

The idea behind the distinction of these branches comes from the previous chapters. A thick branch represents a branch that is closest to being "complete-tree-like". It is comparable to the branches of a complete tree (see Figure 5.3 (c)) where there is a positive probability in Player 1's safe strategy only on the first vertex of the branch. On the other hand, we have branches that are closest to being "path-like". These include the branches of paths (see Figure 5.3 (a)) and spiders (see Figure 5.3 (b)) where the positive probabilities in Player 1's safe strategy are distributed along the branch up to a certain distance. Thus, the medium branches will be branches on which we have positive probabilities on the first two vertices with the lowest weights, while the thin branches will be branches on which we include positive probabilities on the first three vertices with the lowest weights. In the case that a branch at the centroid is a path, a better safe strategy for Player 1 is likely to be obtained by including positive probabilities on more than the first three vertices with the lowest weights. However, the possible configurations of the first k vertices with the lowest weights in a branch at the centroid, multiplies as k increases. Thus, in this thesis, we limit ourselves to three vertices. Nevertheless, one should keep in mind that for a specific example, the

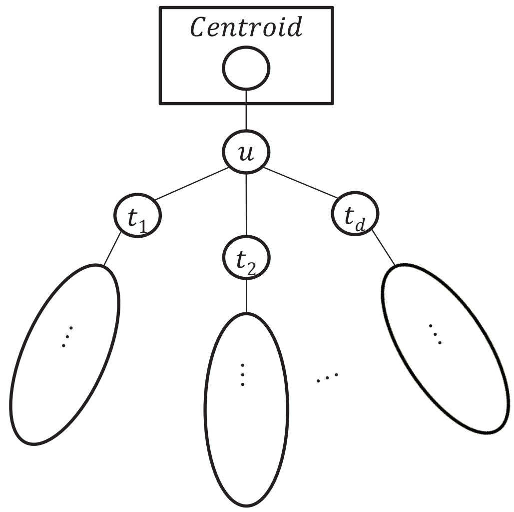


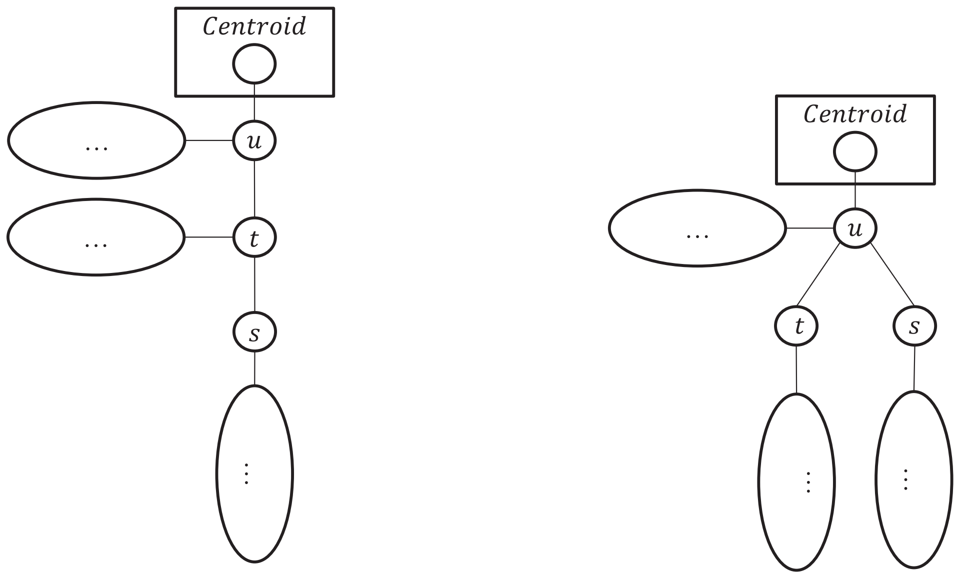
Figure 5.4: Illustration in the proof of Lemma 5.8.

arguments and results that follow could be generalized to include positive probabilities on more vertices in the branches. As we go along, we will discuss the types of trees for which this might be beneficial.

Let us study the possible configurations of the three vertices with the lowest weights in a branch at the centroid.

Lemma 5.8. *Consider a branch at the centroid in a centroidal tree with n vertices, ${}_{C}T(n)$. The vertex with the lowest weight in the branch is the one adjacent to the centroid and the vertex with the second lowest weight is a vertex adjacent to the first.*

Proof. Let B be a branch at the centroid in ${}_{C}T(n)$ with u as the vertex adjacent to the centroid and the vertices $t_1, t_2, ..., t_d$ adjacent to u (see Figure 5.4). From Lemma 5.2, we know that the weight of a vertex is precisely the number of edges in the branch in which lies the centroid. Thus, u is the vertex in B with the minimal weight since for any other vertex, v, in B, the edges in the branch at u in which lies the centroid are a subset of the edges in the branch at v in which lies the centroid. Now, suppose the vertex with the second lowest weight, v, is not adjacent to u, i.e. not one of the vertices t_i for $1 \leq i \leq d$. Considering the centroid as the root of a rooted tree, we have that v is a descendant of one of the vertices t_i, $1 \leq i \leq d$, say t_k. In this case, the weight of v needs to be larger than t_k since the branch at v in which lies the



(a) First configuration with $w(u) < w(t) \leq w(s)$.

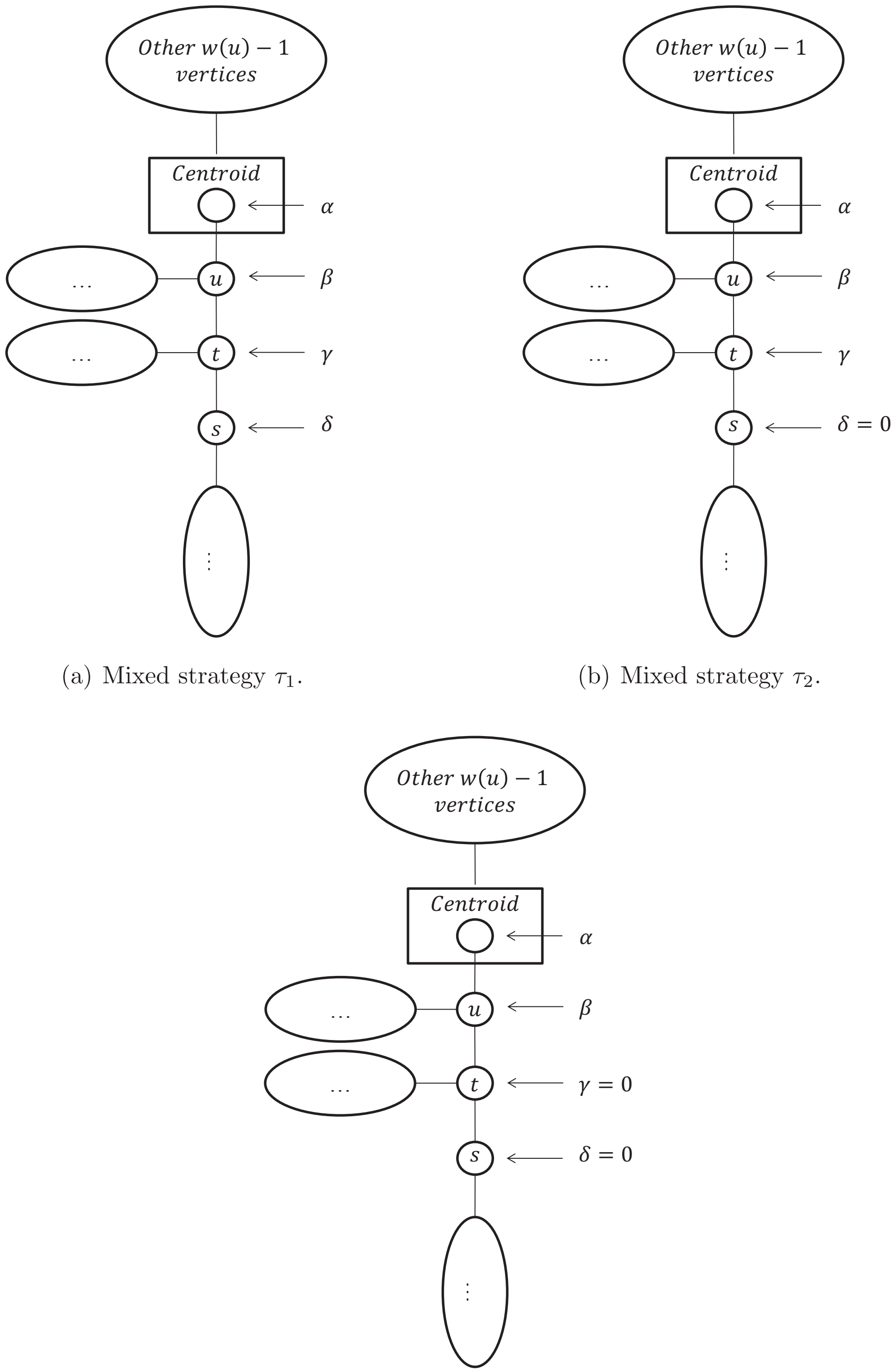
(b) Second configuration with $w(u) < w(t) \leq w(s)$.

Figure 5.5: Possible configurations of the three vertices with lowest weight in a branch at the centroid of a centroidal tree.

centroid also includes the edges in the branch at t_k in which lies the centroid. This is a contradiction to the fact that v is the vertex with the second lowest weight in B. Thus, v needs to be adjacent to u. □

From Lemma 5.8, we know that the vertex with the lowest weight is adjacent to the centroid and a vertex with the second lowest weight is adjacent to the first vertex. That being said, there are two possibilities for the the next vertex with the lowest weight i.e. the next vertex with the lowest weight can be a vertex adjacent to either the first or second vertex with lowest weight in the branch (see Figure 5.5.)

Consider mixed strategies for Player 1 where she chooses with a positive probability only the centroid and some of the three vertices with the lowest weights in a branch at the centroid. If the branch has the configuration (a) of Figure 5.5, we suggest three possible strategies for Player 1, τ_1, τ_2 and τ_3 (see Figure 5.6). The strategy τ_1, represented in Figure 5.6 (a) has Player 1 assigning a probability α on the centroid, a probability β on the vertex with the lowest weight, u, a probability

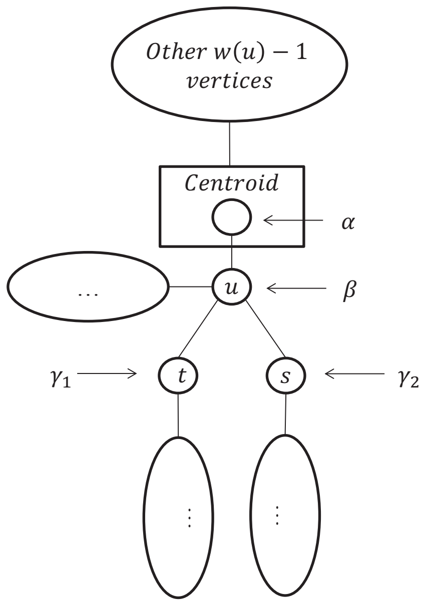


(a) Mixed strategy τ_1.

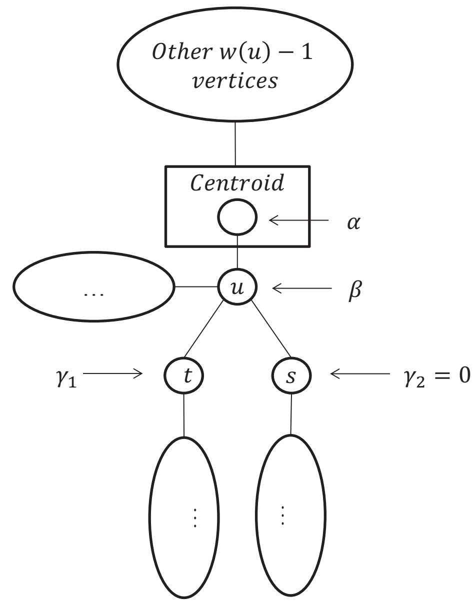
(b) Mixed strategy τ_2.

(c) Mixed strategy τ_3.

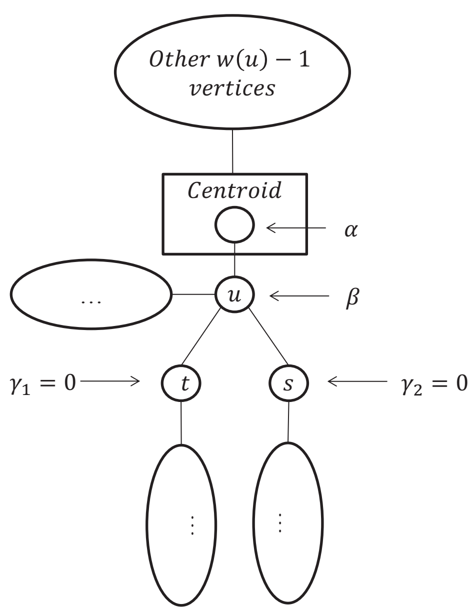
Figure 5.6: Safe strategies of Player 1 on a branch of a centroidal tree with configuration (a) of Figure 5.5.



(a) Mixed strategy τ_4.



(b) Mixed strategy τ_2.



(c) Mixed strategy τ_3.

Figure 5.7: Safe strategies of Player 1 on a branch of a centroidal tree with configuration (b) of Figure 5.5.

γ on the vertex with the second lowest weight, t, a probability δ on the next vertex with the lowest weight, s, and a probability zero on all the other vertices. Following a similar idea as with the strategy $C_{S2}(k)$ for the spiders and the strategies μ_1 and μ_2 for the complete trees, we can determine the expected gains of Player 1 if Player 2 chooses the centroid, the vertex u, the vertex t and the vertex s and force these expected gains to be equal by solving for α, β, γ and δ, knowing that $\alpha+\beta+\gamma+\delta = 1$. This leads to the following mixed strategy and expected gain.

Definition 5.9. Let τ_1 be a mixed strategy on a centroidal tree ${}_CT(n)$ with centroid vertex c and a branch at the centroid, B, with the configuration of Figure 5.6 (a). We define $\tau_1 = (x_1, x_2, ..., x_n)$ where x_i, the probability of choosing the vertex v_i, is

$$x_i = \begin{cases} \alpha, & \text{if } v_i = c \\ \beta = \left(\frac{\overline{w}(t)(w(u)\overline{w}(s)+(w(t)-w(s))(w(t)-w(u)))}{\overline{w}(s)\overline{w}(u)\overline{w}(t)+w(s)w(t)(w(s)-w(t))}\right)\alpha, & \text{if } v_i = u \\ \gamma = \left(\frac{w(t)}{\overline{w}(t)}\right)\beta, & \text{if } v_i = t \\ \delta = \left(\frac{w(s)}{\overline{w}(s)}\right)\gamma + \left(\frac{w(t)-w(u)}{\overline{w}(s)}\right)\alpha, & \text{if } v_i = s \\ 0, & \text{otherwise.} \end{cases} \tag{5.13}$$

with α, β, γ and δ known by solving $\alpha + \beta + \gamma + \delta = 1$ and where u, t and s are respectively the first, second and third vertices with the lowest weights in B.

Lemma 5.10. *Consider the two-player game of Competitive Diffusion on a centroidal tree ${}_CT(n)$ with centroid vertex c and a branch at the centroid, B, with the configuration of Figure 5.6 (a). The expected gains of Player 1 when she chooses the mixed strategy τ_1 and Player 2 chooses the mixed strategies $Z(c)$, $Z(u)$, $Z(t)$ and $Z(s)$ are equal and are*

$$\frac{\overline{w}(t)(w(t)(n^2 - w(u)w(t) + w(t)^2 - w(s)(\overline{w}(u) - w(u) + w(t)))}{n^3 - n^2(w(u) + w(s)) + nw(u)(w(u) + w(s)) - w(t)(w(u) - w(s))(w(u) - w(t) + w(s))} \tag{5.14}$$

where u, t and s are respectively the first, second and third vertices with the lowest weights in B.

Proof. The expected gain of Player 1 with the strategy τ_1 on the centroidal tree ${}_CT(n)$

if Player 2 chooses a vertex v_j for $1 \leq j \leq n$ is

$$\begin{aligned} Gain({}_C T(n),\tau_1, Z(v_j)) = \alpha \cdot Gain({}_C T(n), Z(c), Z(v_j)) \\ + \beta \cdot Gain({}_C T(n), Z(u), Z(v_j)) + \gamma \cdot Gain({}_C T(n), Z(t), Z(v_j)) \\ + \delta \cdot Gain({}_C T(n), Z(s), Z(v_j)) \end{aligned} \tag{5.15}$$

where α, β, γ and δ are defined as in Definition 5.9.

(a) If $v_j = c$, i.e. Player 2 chooses the centroid, then (5.15) becomes

$$Gain({}_C T(n), \tau_1, Z(c)) = \alpha \cdot 0 + \beta \cdot \overline{w}(u) + \gamma \cdot \overline{w}(t) + \delta \cdot \overline{w}(t) \tag{5.16}$$

Replacing the values of α, β, γ and δ known from solving $\alpha + \beta + \gamma + \delta = 1$ gives

$$Gain({}_C T(n), \tau_1, Z(c)) = \frac{\overline{w}(t)(w(t)(n^2 - w(u)w(t) + w(t)^2 - w(s)(\overline{w}(u) - w(u) + w(t)))}{n^3 - n^2(w(u) + w(s)) + nw(u)(w(u) + w(s)) - w(t)(w(u) - w(s))(w(u) - w(t) + w(s))}. \tag{5.17}$$

(b) If $v_j = u$, i.e. Player 2 chooses the vertex with the lowest weight in the branch B, (5.15) becomes

$$Gain({}_C T(n), \tau_1, Z(u)) = \alpha \cdot w(u) + \beta \cdot 0 + \gamma \cdot \overline{w}(t) + \delta \cdot \overline{w}(s) \tag{5.18}$$

Replacing the values of α, β, γ and δ know from solving $\alpha + \beta + \gamma + \delta = 1$ gives

$$Gain({}_C T(n), \tau_1, Z(u)) = \frac{\overline{w}(t)(w(t)(n^2 - w(u)w(t) + w(t)^2 - w(s)(\overline{w}(u) - w(u) + w(t)))}{n^3 - n^2(w(u) + w(s)) + nw(u)(w(u) + w(s)) - w(t)(w(u) - w(s))(w(u) - w(t) + w(s))}. \tag{5.19}$$

(c) If $v_j = t$, i.e. Player 2 chooses a vertex with the second lowest weight in the branch B, (5.15) becomes

$$Gain({}_C T(n), \tau_1, Z(t)) = \alpha \cdot w(u) + \beta \cdot w(t) + \gamma \cdot 0 + \delta \cdot \overline{w}(s) \tag{5.20}$$

Replacing the values of α, β, γ and δ know from solving $\alpha + \beta + \gamma + \delta = 1$ gives

$$Gain({}_CT(n), \tau_1, Z(t)) = \frac{\overline{w}(t)(w(t)(n^2 - w(u)w(t) + w(t)^2 - w(s)(\overline{w}(u) - w(u) + w(t)))}{n^3 - n^2(w(u) + w(s)) + nw(u)(w(u) + w(s)) - w(t)(w(u) - w(s))(w(u) - w(t) + w(s))}. \tag{5.21}$$

(d) If $v_j = s$, i.e. Player 2 chooses the third vertex with the lowest weight in the branch B, (5.15) becomes

$$Gain({}_CT(n), \tau_1, Z(s)) = \alpha \cdot w(t) + \beta \cdot w(t) + \gamma \cdot w(s) + \delta \cdot 0 \tag{5.22}$$

Replacing the values of α, β, γ and δ know from solving $\alpha + \beta + \gamma + \delta = 1$ gives

$$Gain({}_CT(n), \tau_1, Z(s)) = \frac{\overline{w}(t)(w(t)(n^2 - w(u)w(t) + w(t)^2 - w(s)(\overline{w}(u) - w(u) + w(t))w(s))}{n^3 - n^2(w(u) + w(s)) + nw(u)(w(u) + w(s)) - w(t)(w(u) - w(s))(w(u) - w(t) + w(s))}. \tag{5.23}$$

□

In a similar manner, we define the mixed strategies τ_2 and τ_3 of Figure 5.6 (b) and (c) with their associated expected gains.

Definition 5.11. Let τ_2 be a mixed strategy on a centroidal tree ${}_CT(n)$ with centroid vertex c and a branch at the centroid, B, with the configuration of Figure 5.6 (b). We define $\tau_2 = (x_1, x_2, ..., x_n)$ where x_i, the probability of choosing the vertex v_i, is

$$x_i = \begin{cases} \alpha, & \text{if } v_i = c \\ \beta = \left(\frac{w(u)}{\overline{w}(u)}\right)\alpha, & \text{if } v_i = u \\ \gamma = \left(\frac{w(t)}{\overline{w}(t)}\right)\beta, & \text{if } v_i = t \\ \delta = 0, & \text{if } v_i = s \\ 0, & \text{otherwise.} \end{cases} \tag{5.24}$$

with α, β and γ known by solving $\alpha + \beta + \gamma = 1$ and where u, t and s are respectively the first, second and third vertices with the lowest weights in B.

Lemma 5.12. *Consider the two-player game of Competitive Diffusion on a centroidal tree ${}_CT(n)$ with centroid vertex c and a branch at the centroid, B, with the configuration of Figure 5.6 (b). The expected gains of Player 1 when she chooses the mixed* *strategy* τ_2 *and Player 2 chooses the mixed strategies* $Z(c)$, $Z(u)$ *and* $Z(t)$ *are equal and are*

$$\frac{w(u)\overline{w}(u)\overline{w}(t) + w(u)w(t)\overline{w}(t)}{n^2 - nw(t) + w(u)w(t)} \tag{5.25}$$

where u *and* t *are respectively the first and second vertices with the lowest weights in* B.

Proof. This proof follows the same ideas as the proof of Lemma 5.10. □

Definition 5.13. Let τ_3 be a mixed strategy on a centroidal tree ${}_CT(n)$ with centroid vertex c and a branch at the centroid, B, with the configuration of Figure 5.6 (c). We define $\tau_3 = (x_1, x_2, ..., x_n)$ where x_i, the probability of choosing the vertex v_i, is

$$x_i = \begin{cases} \alpha, & \text{if } v_i = c \\ \beta = \left(\frac{w(u)}{\overline{w}(u)}\right)\alpha, & \text{if } v_i = u \\ \gamma = 0, & \text{if } v_i = t \\ \delta = 0, & \text{if } v_i = s \\ 0, & \text{otherwise.} \end{cases} \tag{5.26}$$

with α and β known by solving $\alpha + \beta = 1$ and where u, t and s are respectively the first, second and third vertices with the lowest weights in B.

Lemma 5.14. *Consider the two-player game of Competitive Diffusion on a centroidal tree* ${}_CT(n)$ *with centroid vertex* c *and a branch at the centroid,* B, *with the configuration of Figure 5.6 (c). The expected gains of Player 1 when she chooses the mixed strategy* τ_3 *and Player 2 chooses the mixed strategies* $Z(c)$ *and* $Z(u)$ *are equal and are*

$$\frac{w(u)\overline{w}(u)}{n} \tag{5.27}$$

where u *is the vertex with the lowest weight in* B.

Proof. This proof follows the same ideas as the proof of Lemma 5.10. □

On the other hand, if a branch at the centroid has the configuration (b) of Figure 5.5, we also suggest three possible mixed strategies for Player 1. However, two of them are equivalent to ones already defined (see Figure 5.7). Let us define the new strategy τ_4 obtained in a similar manner as the others.

Definition 5.15. Let τ_4 be a mixed strategy on a centroidal tree ${}_CT(n)$ with centroid vertex c and a branch at the centroid, B, with the configuration of Figure 5.7 (a). We define $\tau_4 = (x_1, x_2, ..., x_n)$ where x_i, the probability of choosing the vertex v_i, is

$$x_i = \begin{cases} \alpha, & \text{if } v_i = c \\ \beta = \left(\frac{w(u)}{\overline{w}(u)}\right)\alpha, & \text{if } v_i = u \\ \gamma_1 = \left(\frac{w(t)}{\overline{w}(t)}\right)\beta, & \text{if } v_i = t \\ \gamma_2 = \left(\frac{w(s)}{\overline{w}(s)}\right)\beta, & \text{if } v_i = s \\ 0, & \text{otherwise.} \end{cases} \tag{5.28}$$

with α, β, γ_1 and γ_2 known by solving $\alpha + \beta + \gamma_1 + \gamma_2 = 1$ and where u, t and s are respectively the first, second and third vertices with the lowest weights in B.

Lemma 5.16. *Consider the two-player game of Competitive Diffusion on a centroidal tree ${}_CT(n)$ with centroid vertex c and a branch at the centroid, B, with the configuration of Figure 5.7 (a). The expected gains of Player 1 when she chooses the mixed strategy τ_4 and Player 2 chooses the mixed strategies $Z(c)$, $Z(u)$, $Z(t)$ and $Z(s)$ are equal and are*

$$\frac{w(u)\overline{w}(t)\overline{w}(s)(\overline{w}(u) + w(t) + w(s))}{n^3 - 2w(u)w(t)w(s) - n^2(w(t) + w(s)) + n(w(t)w(s) + w(u)w(t) + w(u)w(s))} \tag{5.29}$$

where u, t and s are respectively the first, second and third vertices with the lowest weights in B.

Proof. This proof follows the same ideas as the proof of Lemma 5.10. □

By comparing the expected gains (5.14), (5.25), (5.27) and (5.29) we can determine the following:

- The expected gain with the strategy τ_3 (5.27) is always greater than the expected gain with the strategy τ_4 (5.29).

- The expected gain with the strategy τ_2 (5.25) is greater than the expected gain with the strategy τ_3 (5.27) if $w(t) < \overline{w}(u) + \frac{w(u)^2}{n}$.

- The expected gain with the strategy τ_1 (5.14) is greater than the expected gain with the strategy τ_2 (5.25) if $w(t) < \overline{w}(u) + \frac{w(u)^2}{n}$ and $w(s) < \overline{w}(t) + \frac{w(t)^2+(w(t)-w(u))^2}{n+(w(t)-w(u))}$.

In other words, on a thin branch, the expected gain of Player 1 is greater with the strategy τ_1, on a medium branch, the expected gain of Player 1 is greater with the strategy τ_2 while on a thick branch, the expected gain of Player 1 is greater with the strategy τ_3. This suggests a safe strategy for Player 1 which has positive probabilities on some of the branches at the centroid and in which the distribution of probabilities are defined dependently on the branches being thick, medium or thin.

5.2.2 Centroidal Safe Strategy (CSS) Algorithm

To determine the branches on which there should be a positive probability in the safe strategy of Player 1, we consider a stepwise algorithm. The algorithm starts by having positive probabilities on one of the branches at the centroid and adds positive probabilities on other branches one at a time. Branches are added until the expected gain of Player 1 when Player 2 chooses the centroid can no longer be increased by adding some positive probabilities on the remaining branches. Thus, the algorithm consists of two important elements: the order in which the branches should be included and the time when the algorithm should stop adding positive probabilities on some new branches. On that note, we have the following lemmas.

Lemma 5.17. *Consider a centroidal tree ${}_{C}T(n)$ with vertices $\{v_1, v_2, ..., v_n\}$. Suppose we have a mixed strategy for Player 1 on ${}_{C}T(n)$, $\sigma = (x_1, x_2, ..., x_n)$ where x_i is the probability of choosing the vertex v_i and specifically*

$$x_i = \begin{cases} \alpha, & \text{if } v_i = c \\ 0, & \text{if } v_i \in B \end{cases} \tag{5.30}$$

with $0 < \alpha < 1$ and B, a thick branch at the centroid. Suppose we add probabilities in terms of α on the vertices of B as described in (5.26). That is, we form a vector

$Q = (q_1, q_2, ..., q_n)$ such that

$$q_i = \begin{cases} \beta = \left(\frac{w(u)}{\overline{w}(u)}\right)\alpha, & \text{if } v_i = u \\ 0, & \text{if } v_i \in B \text{ and } v_i \neq u \\ x_i, & \text{if } v_i \notin B \end{cases} \tag{5.31}$$

where u is the vertex with the lowest weight in B. Let $\sigma' = (x'_1, x'_2, ..., x'_n)$ be a scale down of the vector Q where,

$$x'_i = \frac{q_i}{\sum_{i=1}^{n} q_i}. \tag{5.32}$$

Consider σ' as a new mixed strategy for Player 1. Then, we have

$$Gain({}_CT(n), \sigma', Z(c)) \geq Gain({}_CT(n), \sigma, Z(c)) \tag{5.33}$$

if and only if

$$\overline{w}(u) \geq Gain({}_CT(n), \sigma, Z(c)). \tag{5.34}$$

Moreover,

$$Gain({}_CT(n), \sigma', Z(c)) \leq \overline{w}(u) \tag{5.35}$$

that is, the new expected gain does not surpass $\overline{w}(u)$.

Proof. Suppose the centroid c corresponds to the vertex v_1 of ${}_CT(n)$. The expected gain of Player 1 with the strategy σ when Player 2 chooses the centroid is

$$Gain({}_CT(n), \sigma, Z(c)) = \alpha \cdot 0 + \sum_{i=2}^{n} x_i \cdot Gain({}_CT(n), Z(v_i), Z(c)) \tag{5.36}$$

Now, suppose there are m vertices in the branch B and that they correspond to the vertices $\{v_2, v_3, ..., v_{m+1}\}$ of ${}_CT(n)$. Let the vertex v_2 be specifically the vertex u. The expected gain of Player 1 with the strategy σ' when Player 2 chooses the centroid is

$$Gain({}_CT(n), \sigma', Z(c)) = \sum_{i=1}^{n} x'_i \cdot Gain({}_CT(n), Z(v_i), Z(c)). \tag{5.37}$$

Replacing the values of x'_i from (5.32) gives

$$Gain({}_CT(n), \sigma', Z(c)) = \sum_{i=1}^{n} \frac{q_i}{\sum_{i=1}^{n} q_i} \cdot Gain({}_CT(n), Z(v_i), Z(c)). \tag{5.38}$$

Since $q_i = x_i$ for $1 \leq i \leq n$, $i \neq 2$ and $q_2 = \frac{w(u)}{\overline{w}(u)}$, we have

$$Gain({}_CT(n), \sigma', Z(c)) = \frac{1}{\sum_{i=1}^{n} q_i} \left(\alpha \cdot 0 + \left(\frac{w(u)}{\overline{w}(u)} \right) \cdot Gain({}_CT(n), Z(u), Z(c)) + \sum_{i=3}^{n} x_i \cdot Gain({}_CT(n), Z(v_i), Z(c)) \right). \tag{5.39}$$

Moreover, $x_2 = 0$, since initially, there was no positive probabilities on the vertices of the branch B. Thus,

$$Gain({}_CT(n), \sigma', Z(c)) = \frac{1}{\sum_{i=1}^{n} q_i} \left(\left(\frac{w(u)}{\overline{w}(u)} \right) \cdot Gain({}_CT(n), Z(u), Z(c)) + Gain({}_CT(n), \sigma, Z(c)) \right). \tag{5.40}$$

Therefore,

$$\begin{aligned} & Gain({}_CT(n), \sigma', Z(c)) \geq Gain({}_CT(n), \sigma, Z(c)) \\ \Leftrightarrow & \frac{1}{\sum_{i=1}^{n} q_i} \left(\left(\frac{w(u)}{\overline{w}(u)} \right) \cdot Gain({}_CT(n), Z(u), Z(c)) \right) \geq \\ & \left(\frac{\sum_{i=1}^{n} q_i - 1}{\sum_{i=1}^{n} q_i} \right) Gain({}_CT(n), \sigma, Z(c)). \end{aligned} \tag{5.41}$$

This can be simplified to

$$Gain({}_CT(n), Z(u), Z(c)) \geq Gain({}_CT(n), \sigma, Z(c)) \tag{5.42}$$

since $\sum_{i=1}^{n} q_i - 1 = \frac{w(u)}{\overline{w}(u)}$. Finally, recall from Lemma 5.2, that $w(u)$ is the number of edges in the branch at u in which lies the centroid. Thus, there are $\overline{w}(u)$ vertices in the branch B and so $Gain({}_CT(n), Z(u), Z(c)) = \overline{w}(u)$.

Lastly, if $Gain({}_CT(n), Z(u), Z(c)) \geq Gain({}_CT(n), \sigma, Z(c))$, we can deduce from (5.40) that

$$Gain({}_CT(n), \sigma', Z(c)) \leq Gain({}_CT(n), Z(u), Z(c)) = \overline{w}(u) \tag{5.43}$$

since

$$0 < \frac{1}{\sum_{i=1}^{n} q_i} \left(\frac{w(u)}{\overline{w}(u)} \right) < 1 \text{ and } 0 < \frac{1}{\sum_{i=1}^{n} q_i} < 1. \tag{5.44}$$

□

Lemma 5.18. *Consider a centroidal tree ${}_CT(n)$ with vertices $\{v_1, v_2, ..., v_n\}$. Suppose we have a mixed strategy for Player 1 on ${}_CT(n)$, $\sigma = (x_1, x_2, ..., x_n)$, where x_i is the probability of choosing the vertex v_i and specifically*

$$x_i = \begin{cases} \alpha, & \text{if } v_i = c \\ 0, & \text{if } v_i \in B \end{cases} \tag{5.45}$$

with $0 < \alpha < 1$ and B, a medium branch at the centroid. Suppose we add probabilities in terms of α on the vertices of B as described in (5.24). That is, we have a vector $Q = (q_1, q_2, ..., q_n)$ such that

$$q_i = \begin{cases} \beta = \left(\frac{w(u)}{\overline{w}(u)}\right)\alpha, & \text{if } v_i = u \\ \gamma = \left(\frac{w(t)}{\overline{w}(t)}\right)\beta, & \text{if } v_i = t \\ 0, & \text{if } v_i \in B \text{ and } v_i \neq u, t \\ x_i, & \text{if } v_i \notin B \end{cases} \tag{5.46}$$

where u and t are the first and second vertices with the lowest weights in B. Let $\sigma' = (x'_1, x'_2, ..., x'_n)$ be a scale down of the vector Q where,

$$x'_i = \frac{q_i}{\sum_{i=1}^{n} q_i}. \tag{5.47}$$

Consider σ' as a new mixed strategy for Player 1. Then, we have

$$Gain({}_CT(n), \sigma', Z(c)) \geq Gain({}_CT(n), \sigma, Z(c)) \tag{5.48}$$

if and only if

$$\left(\frac{\overline{w}(t)}{n}\right)\overline{w}(u) + \left(\frac{w(t)}{n}\right)\overline{w}(t) \geq Gain({}_CT(n), \sigma, Z(c)). \tag{5.49}$$

Moreover,

$$Gain({}_CT(n), \sigma', Z(c)) \leq \left(\frac{\overline{w}(t)}{n}\right)\overline{w}(u) + \left(\frac{w(t)}{n}\right)\overline{w}(t). \tag{5.50}$$

Proof. This proof follows the same ideas as the proof of Lemma 5.17. □

Lemma 5.19. *Consider a centroidal tree ${}_CT(n)$ with vertices $\{v_1, v_2, ..., v_n\}$. Suppose we have a mixed strategy for Player 1 on ${}_CT(n)$, $\sigma = (x_1, x_2, ..., x_n)$, where x_i is the probability of choosing the vertex v_i and specifically*

$$x_i = \begin{cases} \alpha, & \text{if } v_i = c \\ 0, & \text{if } v_i \in B \end{cases} \tag{5.51}$$

with $0 < \alpha < 1$ *and* B*, a thin branch at the centroid. Suppose we add probabilities in terms of* α *on the vertices of* B *as described in (5.13). That is, we have a vector* $Q = (q_1, q_2, ..., q_n)$ *such that*

$$q_i = \begin{cases} \beta = \left(\frac{\overline{w}(t)(w(u)\overline{w}(s)+(w(t)-w(s))(w(t)-w(u)))}{\overline{w}(s)\overline{w}(u)\overline{w}(t)+w(s)w(t)(w(s)-w(t))}\right)\alpha, & \text{if } v_i = u \\ \gamma = \left(\frac{w(t)}{\overline{w}(t)}\right)\beta, & \text{if } v_i = t \\ \delta = \left(\frac{w(s)}{\overline{w}(s)}\right)\gamma + \left(\frac{w(t)-w(u)}{\overline{w}(s)}\right)\alpha, & \text{if } v_i = s \\ 0, & \text{if } v_i \in B \text{ and } v_i \neq u,\ t,\ s \\ x_i, & \text{if } v_i \notin B \end{cases} \tag{5.52}$$

where u*,* t *and* s *are respectively the first, second and third vertices with the lowest weights in B. Let* $\sigma' = (x'_1, x'_2, ..., x'_n)$ *be a scale down of the vector* Q *where,*

$$x'_i = \frac{q_i}{\sum_{i=1}^{n} q_i}. \tag{5.53}$$

Consider σ' *as a new strategy for Player 1. We have*

$$Gain({}_CT(n), \sigma', Z(c)) \geq Gain({}_CT(n), \sigma, Z(c)) \tag{5.54}$$

if and only if

$$\begin{aligned} &\frac{w(t)\overline{w}(t)(n^2 - nw(s) - w(s)w(t) + w(t)^2 + 2w(s)w(u) - w(t)w(u))}{nw(t)\overline{w}(s) + w(u)w(t)(-n + w(s) + w(t)) + \overline{w}(t)w(u)^2} \\ &\geq Gain({}_CT(n), \sigma, Z(c)). \end{aligned} \tag{5.55}$$

Moreover,

$$\begin{aligned} &Gain({}_CT(n), \sigma', Z(c)) \leq \\ &\frac{w(t)\overline{w}(t)(n^2 - nw(s) - w(s)w(t) + w(t)^2 + 2w(s)w(u) - w(t)w(u))}{nw(t)\overline{w}(s) + w(u)w(t)(-n + w(s) + w(t)) + \overline{w}(t)w(u)^2}. \end{aligned} \tag{5.56}$$

Proof. This proof follows the same ideas as the proof of Lemma 5.17. □

We now have conditions that permit us to determine if adding positive probabilities on the vertices of a branch increases the expected gain of Player 1 when Player 2 chooses the centroid. However, we must decide which branch should be added first if two or more branches satisfy the conditions. To do this, we first define the criterion of a branch. The criterion of a branch is dependant of the type of branch and corresponds respectively to the equations (5.34), (5.49) and (5.55) in the Lemmas 5.17, 5.18 and 5.19. Thus, we will have that the expected gain of Player 1 can be increased by adding positive probabilities on the vertices of a branch if and only if its criterion is greater than the current expected gain.

Definition 5.20. For a branch at the centroid B in a centroidal tree ${}_CT(n)$, we define the **criterion** of B, $Cr(B)$, as

$$Cr(B) = \begin{cases} \overline{w}(u), & \text{if } B \text{ is a thick branch} \\ \left(\frac{\overline{w}(t)}{n}\right)\overline{w}(u) + \left(\frac{w(t)}{n}\right)\overline{w}(t), & \text{if } B \text{ is a medium branch} \\ \frac{w(t)\overline{w}(t)(n^2 - nw(s) - w(s)w(t) + w(t)^2 + 2w(s)w(u) - w(t)w(u))}{nw(t)\overline{w}(s) + w(u)w(t)(-n + w(s) + w(t)) + \overline{w}(t)w(u)^2}, & \text{if } B \text{ is a thin branch} \end{cases} \tag{5.57}$$

where u is the vertex with the lowest weight in the branch B, t is the vertex with the second lowest weight in the branch B and s is the next vertex with the lowest weight in the branch B. By assumption, a branch that has less than three vertices will be assigned a criterion of 0.

We now argue the ordering of the branches in the algorithm. Let B_i and B_j be two branches at the centroid in a centroidal tree ${}_CT(n)$ with vertices $\{v_1, v_2, ..., v_n\}$. Suppose we have a mixed strategy for Player 1 on ${}_CT(n)$, $\sigma = (x_1, x_2, ..., x_n)$, where x_k, the probability of choosing the vertex v_k, is

$$x_k = \begin{cases} \alpha, & \text{if } v_k = c \\ 0, & \text{if } v_k \in B_i \text{ or } v_k \in B_j \\ \epsilon_k \cdot \alpha, & \text{for } v_k \text{ otherwise} \end{cases} \tag{5.58}$$

where α is such that $\alpha + \sum_k \epsilon_k \cdot \alpha = 1$ and $0 \leq \epsilon_k \leq 1$. The expected gain of Player 1 with the strategy σ when Player 2 chooses the centroid is $Gain({}_CT(n), \sigma, Z(c))$. Suppose $Cr(B_i) \geq Cr(B_j) \geq Gain({}_CT(n), \sigma, Z(c))$. Since the criterion of both branches is greater than the current expected gain, we know by Lemmas 5.17, 5.18 and 5.19, that adding some positive probabilities on either branches will increase the expected gain of Player 1. Thus, we have two scenarios.

(i) If we start by adding some probabilities on the branch B_j to get the mixed strategy $\sigma_j = (y_1, y_2, ..., y_n)$ where, y_k, the probability of choosing the vertex v_k is:

(a) If B_j is a thin branch

$$y_k = \begin{cases} \alpha, & \text{if } v_k = c \\ 0, & \text{if } v_k \in B_i \\ \beta_j = \left(\frac{\overline{w}(t_j)(w(u_j)\overline{w}(s_j)+(w(t_j)-w(s_j))(w(t_j)-w(u_j)))}{\overline{w}(s_j)\overline{w}(u_j)\overline{w}(t_j)+w(s_j)w(t_j)(w(s_j)-w(t_j))} \right) \alpha, & \text{if } v_k = u_j \in B_j \\ \gamma_j = \left(\frac{w(t_j)}{\overline{w}(t_j)} \right) \beta_j, & \text{if } v_k = t_j \in B_j \\ \delta_j = \left(\frac{w(s_j)}{\overline{w}(s_j)} \right) \gamma_j + \left(\frac{w(t_j)-w(u_j)}{\overline{w}(s_j)} \right) \alpha, & \text{if } v_k = s_j \in B_j \\ 0, & \text{if } v_k \in B_j, v_k \neq u_j, t_j, s_j \\ \epsilon_k \cdot \alpha, & \text{for } v_k \text{ otherwise.} \end{cases} \tag{5.59}$$

with α such that $\alpha + \beta_j + \gamma_j + \delta_j + \sum_k \epsilon_k \cdot \alpha = 1$, where the ϵ_k's are unchanged and where u_j, t_j and s_j are respectively the vertices with the first, second and third lowest weights in B_j. The probabilities on the branch B_j are as in (5.13) of the strategy τ_1.

(b) If B_j is a medium branch

$$y_k = \begin{cases} \alpha, & \text{if } v_k = c \\ 0, & \text{if } v_k \in B_i \\ \beta_j = \left(\frac{w(u_j)}{\overline{w}(u_j)} \right) \alpha, & \text{if } v_k = u_j \in B_j \\ \gamma_j = \left(\frac{w(t_j)}{\overline{w}(t_j)} \right) \beta_j, & \text{if } v_k = t_j \in B_j \\ 0, & \text{if } v_k \in B_j, v_k \neq u_j, t_j \\ \epsilon_k \cdot \alpha, & \text{for } v_k \text{ otherwise.} \end{cases} \tag{5.60}$$

with α such that $\alpha + \beta_j + \gamma_j + \sum_k \epsilon_k \cdot \alpha = 1$, where the ϵ_k's are unchanged and where u_j and t_j are respectively the first and second vertices with lowest weights in B_j. The probabilities on the branch B_j are as in (5.24) of the strategy τ_2.

(c) If B_j is a thick branch

$$y_k = \begin{cases} \alpha, & \text{if } v_k = c \\ 0, & \text{if } v_k \in B_i \\ \beta_j = \left(\frac{w(u_j)}{\overline{w}(u_j)}\right)\alpha, & \text{if } v_k = u_j \in B_j \\ 0, & \text{if } v_k \in B_j, v_k \neq u_j \\ \epsilon_k \cdot \alpha, & \text{for } v_k \text{ otherwise.} \end{cases} \tag{5.61}$$

with α such that $\alpha + \beta_j + \sum_k \epsilon_k \cdot \alpha = 1$, where the ϵ_k's are unchanged and where u_j is the vertex with the lowest weight in B_j. The probabilities on the branch B_j are as described in (5.26) of the strategy τ_3.

The new expected gain of Player 1 when Player 2 chooses the centroid is $Gain({}_CT(n), \sigma_j, Z(c))$.

(ii) On the other hand, if we start by adding some probabilities on the branch B_i to get the mixed strategy $\sigma_i = (z_1, z_2, ..., z_n)$ where, z_k, the probability of choosing the vertex v_k is:

(a) If B_i is a thin branch

$$z_k = \begin{cases} \alpha, & \text{if } v_k = c \\ 0, & \text{if } v_k \in B_j \\ \beta_i = \left(\frac{\overline{w}(t_i)(w(u_i)\overline{w}(s_i)+(w(t_i)-w(s_i))(w(t_i)-w(u_i)))}{\overline{w}(s_i)\overline{w}(u_i)\overline{w}(t_i)+w(s_i)w(t_i)(w(s_i)-w(t_i))}\right)\alpha, & \text{if } v_k = u_i \in B_i \\ \gamma_i = \left(\frac{w(t_i)}{\overline{w}(t_i)}\right)\beta_i, & \text{if } v_k = t_i \in B_i \\ \delta_i = \left(\frac{w(s_i)}{\overline{w}(s_i)}\right)\gamma_i + \left(\frac{w(t_i)-w(u_i)}{\overline{w}(s_i)}\right)\alpha, & \text{if } v_k = s_i \in B_i \\ 0, & \text{if } v_k \in B_i, v_k \neq u_i, t_i, s_i \\ \epsilon_k \cdot \alpha, & \text{for } v_k \text{ otherwise.} \end{cases} \tag{5.62}$$

with α such that $\alpha + \beta_i + \gamma_i + \delta_i + \sum_k \epsilon_k \cdot \alpha = 1$, where the ϵ_k's are unchanged and where u_i, t_i and s_i are respectively the vertices with the first, second and third lowest weights in B_i. The probabilities on the branch B_i are as in (5.13) of the strategy τ_1.

(b) If B_i is a medium branch

$$z_k = \begin{cases} \alpha, & \text{if } v_k = c \\ 0, & \text{if } v_k \in B_j \\ \beta_i = \left(\frac{w(u_i)}{\overline{w}(u_i)}\right)\alpha, & \text{if } v_k = u_i \in B_i \\ \gamma_i = \left(\frac{w(t_i)}{\overline{w}(t_i)}\right), & \text{if } v_k = t_i \in B_i \\ 0, & \text{if } v_k \in B_i, v_k \neq u_i, t_i \\ \epsilon_k \cdot \alpha, & \text{for } v_k \text{ otherwise.} \end{cases} \tag{5.63}$$

with α such that $\alpha + \beta_i + \gamma_i + \sum_k \epsilon_k \cdot \alpha = 1$, where the ϵ_k's are unchanged and where u_i and t_i are respectively the vertices with the first and second lowest weights in B_i. The probabilities on the branch B_i are as in (5.24) of the strategy τ_2.

(c) If B_i is a thick branch

$$z_k = \begin{cases} \alpha, & \text{if } v_k = c \\ 0, & \text{if } v_k \in B_j \\ \beta_i = \left(\frac{w(u_i)}{\overline{w}(u_i)}\right)\alpha, & \text{if } v_k = u_i \in B_i \\ 0, & \text{if } v_k \in B_i, v_k \neq u_i \\ \epsilon_k \cdot \alpha, & \text{for } v_k \text{ otherwise.} \end{cases} \tag{5.64}$$

with α such that $\alpha + \beta_i + \sum_k \epsilon_k \cdot \alpha = 1$, where the ϵ_k's are unchanged and where u_i is the vertex with the lowest weight in B_i. The probabilities on the branch B_i are as in (5.26) of the strategy τ_3.

The new expected gain of Player 1 when Player 2 chooses the centroid is $Gain({}_CT(n), \sigma_i, Z(c))$.

Now, we know by Lemmas 5.17, 5.18 and 5.19 that

$$Gain({}_CT(n), \sigma_j, Z(c)) \leq Cr(B_j)$$

and

$$Gain({}_CT(n), \sigma_i, Z(c)) \leq Cr(B_i).$$

In the first scenario, since $Cr(B_i) \geq Cr(B_j)$, we have $Cr(B_i) \geq Gain(_CT(n), \sigma_j, Z(c))$. Therefore, adding to the strategy σ_j, some positive probabilities on the vertices of the branch B_i and scaling down the probabilities will increase the expected gain of Player 1. In the second scenario, $Cr(B_j)$ might or might not be greater than or equal to $Gain(_CT(n), \sigma_i, Z(c))$. If it is, then adding to the strategy σ_i, some positive probabilities on the vertices of the branch B_j will result in the same final expected gain as in scenario 1. If $Cr(B_j) < Gain(_CT(n), \sigma_i, Z(c))$, then we should not add positive probabilities on the branch B_j since it will decrease the expected gain to Player 1. Thus, the resulting expected gain following the second scenario is either greater or equivalent to the resulting expected gain following the first scenario. For this reason, it is always advantageous to include some positive probabilities on the branch with the largest criterion first.

These observations are summarized in the heuristic algorithm, **Centroidal Safe Strategy (CSS) Algorithm**, represented in Figure 5.8 and defined as follows.

Algorithm 5.21. Centroidal Safe Strategy (CSS) Algorithm

INPUT: Centroidal tree with d branches at the centroid for which all the weights of the vertices are known.

STEP 1: Order the branches $\{B_1, B_2, ..., B_d\}$ such that $Cr(B_i) \geq Cr(B_{i+1})$ for all $1 \leq i \leq d-1$, where $Cr(B_i)$ is the criterion of the branch B_i as defined in Definition 5.20.

STEP 2:

(a) Start with $i = 1$ and consider the branch B_1. Let u_1, t_1, s_1 be the vertices of B_1 such that $w(u_1) < w(t_1) \leq w(s_1) \leq w(r_k)$ for all other vertices r_k in B_1. Form a safe strategy for Player 1, σ_1, in which the centroid is chosen with probability α and the vertices in the branch B_1 are chosen with probabilities that depend on the branch B_1 being thick, medium or thin.

(i) If B_1 is a thin branch, the probabilities are

- $\beta_1 = \left(\frac{\overline{w}(t_1)(w(u_1)\overline{w}(s_1) + (w(t_1) - w(s_1))(w(t_1) - w(u_1)))}{\overline{w}(s_1)\overline{w}(u_1)\overline{w}(t_1) + w(s_1)w(t_1)(w(s_1) - w(t_1))}\right) \alpha$ on u_1
- $\gamma_1 = \left(\frac{w(t_1)}{\overline{w}(t_1)}\right) \beta_1$ on t_1
- $\delta_1 = \left(\frac{w(s_1)}{\overline{w}(s_1)}\right) \gamma_1 + \left(\frac{w(t_1) - w(u_1)}{\overline{w}(s_1)}\right) \alpha$ on s_1
- 0 on the other vertices of B_1.

(5.65)

(ii) If B_1 is a medium branch, the probabilities are

- $\beta_1 = \left(\frac{w(u_1)}{\overline{w}(u_1)}\right) \alpha$ on u_1
- $\gamma_1 = \left(\frac{w(t_1)}{\overline{w}(t_1)}\right) \beta_1$ on t_1
- $\delta_1 = 0$ on s_1
- 0 on the other vertices of B_1.

(5.66)

(iii) If B_1 is a thick branch, the probabilities are

- $\beta_1 = \left(\frac{w(u_1)}{\overline{w}(u_1)}\right) \alpha$ on u_1
- $\gamma_1 = 0$ on t_1
- $\delta_1 = 0$ on s_1
- 0 on the other vertices of B_1.

(5.67)

Skip to STEP 3.

(b) Consider the branch B_i. Let u_i, t_i, s_i be the vertices of B_i such that $w(u_i) < w(t_i) \leq w(s_i) \leq w(r_k)$ for all other vertices r_k in B_i. Form a safe strategy for Player 1, σ_i, in which the centroid is chosen with probability α, the probabilities on the vertices of the branches B_k, $1 \leq k < i$ are the same in terms of α as in the strategy σ_{i-1} and the probabilities on the vertices in the branch B_i depend on the branch B_i being thick, medium or thin.

(i) If B_i is a thin branch, the probabilities are

- $\beta_i = \left(\frac{\overline{w}(t_i)(w(u_i)\overline{w}(s_i) + (w(t_i) - w(s_i))(w(t_i) - w(u_i)))}{\overline{w}(s_i)\overline{w}(u_i)\overline{w}(t_i) + w(s_i)w(t_i)(w(s_i) - w(t_i))}\right) \alpha$ on u_i
- $\gamma_i = \left(\frac{w(t_i)}{\overline{w}(t_i)}\right) \beta_i$ on t_i (5.68)
- $\delta_i = \left(\frac{w(s_i)}{\overline{w}(s_i)}\right) \gamma_i + \left(\frac{w(t_i) - w(u_i)}{\overline{w}(s_i)}\right) \alpha$ on s_i
- 0 on the other vertices of B_i.

(ii) If B_i is a medium branch, the probabilities are

- $\beta_i = \left(\frac{w(u_i)}{\overline{w}(u_i)}\right) \alpha$ on u_i
- $\gamma_i = \left(\frac{w(t_i)}{\overline{w}(t_i)}\right) \beta_i$ on t_i
- $\delta_i = 0$ on s_i
- 0 on the other vertices of B_i.

(5.69)

(iii) If B_i is a thick branch, the probabilities are

- $\beta_i = \left(\frac{w(u_i)}{\overline{w}(u_i)}\right) \alpha$ on u_i
- $\gamma_i = 0$ on t_i
- $\delta_i = 0$ on s_i
- 0 on the other vertices of B_i.

(5.70)

STEP 3: Determine α by solving $\alpha + \sum_{j=1}^{i} (\beta_j + \gamma_j + \delta_j) = 1$ and calculate the expected gain of Player 1 with the strategy σ_i when Player 2 chooses to start with the centroid,

$$Gain({}_CT(n), \sigma_i, Z(c)) = \alpha \cdot 0 + \sum_{j=1}^{i} (\beta_i \cdot \overline{w}(u_i) + \gamma_i \cdot \overline{w}(t_i) + \delta_i \cdot \overline{w}(t_i)) . \quad (5.71)$$

STEP 4:

(a) If $i < d$, verify $Cr(B_{i+1}) \geq Gain({}_CT(n), \sigma_i, Z(c))$.

– If it is true, go to STEP 2 (b) with $i = i+1$. That is, consider another safe strategy for Player 1, σ_{i+1}, where positive probabilities on some vertices in the next branch of the ordering are included.

– If it is false, then the expected gain of Player 1 cannot be increased by adding more branches. The algorithm stops and σ_i is the resulting safe strategy for Player 1.

(b) If $i = d$, there are no more branches at the centroid that can be added. Thus, the algorithm stops and σ_i is the resulting safe strategy for Player 1.

OUTPUT: Safe strategy for Player 1, σ_i, with guaranteed gain

$$GGain({}_CT(n), \sigma_i) = Gain({}_CT(n), \sigma_i, Z(c)). \tag{5.72}$$

Note that in STEP 2 of the algorithm, the probabilities that are added on the vertices of the branch B_i are equivalent to the distribution of the probabilities on a branch at the centroid in the strategies τ_3, τ_2 and τ_1 (see (5.13), (5.24) and (5.26)), respectively when B_i is thin, medium or thick.

As a final point, there is one result from the algorithm left to be proved. The following theorem shows that the expected gain of Player 1 with the strategy σ_k when Player 2 chooses the centroid is in fact the minimal gain that Player 1 can get with the strategy σ_k so we have

$$GGain({}_CT(n), \sigma_k) = Gain({}_cT(n), \sigma_k, Z(c)). \tag{5.73}$$

Theorem 5.22. *Let ${}_CT(n)$ be a centroidal tree with d branches at the centroid. Suppose we apply the algorithm described in Figure 5.8 to ${}_CT(n)$ and we get the mixed strategy σ_k of Player 1 as output. Then,*

$$GGain({}_CT(n), \sigma_k) = Gain({}_CT(n), \sigma_k, Z(c)) \tag{5.74}$$

where c is the centroid of ${}_CT(n)$.

Proof. The mixed strategy σ_k has positive probabilities on vertices of the branches $B_1, B_2, ..., B_k$. From this set of branches, suppose there are k_1 thick branches,

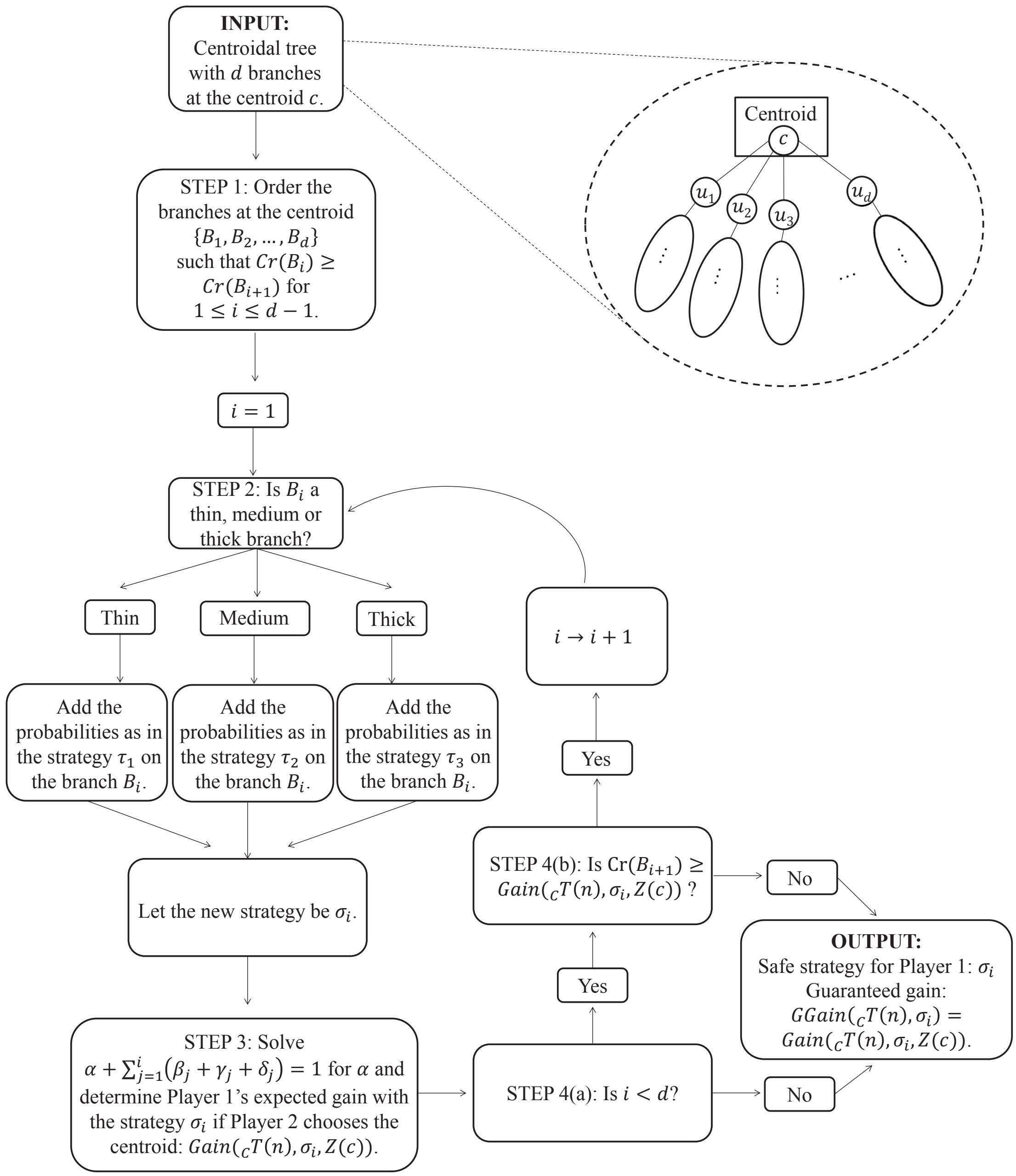


Figure 5.8: Representation of the CSS Algorithm giving a safe strategy for Player 1 along with its guaranteed gain on a centroidal tree ${}_cT(n)$.

$$\{i_1^{Thk}, i_2^{Thk}, ..., i_{k_1}^{Thk}\} \in \{1, 2, 3, ..., k\},$$

k_2 medium branches,

$$\{i_1^{Med}, i_2^{Med}, ..., i_{k_2}^{Med}\} \in \{1, 2, 3, ..., k\},$$

and k_3 thin branches,

$$\{i_1^{Thn}, i_2^{Thn}, ..., i_{k_3}^{Thn}\} \in \{1, 2, 3, ..., k\}$$

such that $k_1 + k_2 + k_3 = k$.

The algorithm defines the strategy σ_k as having the probabilities

$$
\begin{aligned}
&\bullet \quad \alpha \text{ on the centroid } c \\
&\bullet \begin{cases} \beta_i = \left(\frac{w(u_i)}{\overline{w}(u_i)}\right) \alpha \text{ on } u_i \\ \gamma_i = 0 \text{ on } t_i \\ \delta_i = 0 \text{ on } s_i \end{cases} \qquad \text{if } B_i \text{ is a thick branch} \\
&\bullet \begin{cases} \beta_i = \left(\frac{w(u_i)}{\overline{w}(u_i)}\right) \alpha \text{ on } u_i \\ \gamma_i = \left(\frac{w(t_i)}{\overline{w}(t_i)}\right) \beta \text{ on } t_i \\ \delta = 0 \text{ on } s_i \end{cases} \qquad \text{if } B_i \text{ is a medium branch} \\
&\bullet \begin{cases} \beta_i = \left(\frac{\overline{w}(t_i)(w(u_i)\overline{w}(s_i)+(w(t_i)-w(s_i))(w(t_i)-w(u_i)))}{\overline{w}(s_i)\overline{w}(u_i)\overline{w}(t_i)+w(s_i)w(t_i)(w(s_i)-w(t_i))}\right) \alpha \text{ on } u_i \\ \gamma_i = \left(\frac{w(t_i)}{\overline{w}(t_i)}\right) \beta_i \text{ on } t_i \\ \delta_i = \left(\frac{w(s_i)}{\overline{w}(s_i)}\right) \gamma_i + \left(\frac{w(t_i)-w(u_i)}{\overline{w}(s_i)}\right) \alpha \text{ on } s_i \end{cases} \qquad \text{if } B_i \text{ is a thin branch} \\
&\bullet \quad 0 \text{ on all the other vertices}
\end{aligned}
\tag{5.75}
$$

where u_i, t_i and s_i are the vertices of B_i such that $w(u_i) < w(t_i) \leq w(s_i) \leq w(r_i)$ for all other vertices r_i in B_i and where α is such that $\alpha + \sum_{i=1}^{k}(\beta_i + \gamma_i + \delta_i) = 1$.

If Player 2 chooses to start with the centroid c, the expected gain of Player 1 with the strategy σ_k is

$$Gain({}_CT(n), \sigma_k, Z(c)) = \alpha \cdot 0 + \sum_{j=i_1^{Thk}}^{i_{k_1}^{Thk}} \beta_j \cdot \overline{w}(u_j) + \sum_{j=i_1^{Med}}^{i_{k_2}^{Med}} (\beta_j \cdot \overline{w}(u_j) + \gamma_j \cdot \overline{w}(t_j)) + \sum_{j=i_1^{Thn}}^{i_{k_3}^{Thn}} (\beta_j \cdot \overline{w}(u_j) + \gamma_j \cdot \overline{w}(t_j) + \delta_j \cdot \overline{w}(t_j)) . \tag{5.76}$$

If Player 2 chooses to start with a vertex u_r in B_r where B_r is a thin branch, then

$$r \in \{i_1^{Thn}, i_2^{Thn}, ..., i_{k_3}^{Thn}\},$$

and the expected gain of Player 1 with the strategy σ_k is

$$\begin{aligned} Gain({}_CT(n), \sigma_k, Z(u_r)) = \alpha \cdot w(u_r) + \sum_{j=i_1^{Thk}}^{i_{k_1}^{Thk}} \beta_j \cdot \overline{w}(u_j) \\ + \sum_{j=i_1^{Med}}^{i_{k_2}^{Med}} (\beta_j \cdot \overline{w}(u_j) + \gamma_j \cdot \overline{w}(u_j)) \\ + \beta_r \cdot 0 + \gamma_r \cdot \overline{w}(t_r) + \delta_r \cdot \overline{w}(s_r) \\ + \sum_{j=i_1^{Thn},\, j \neq r}^{i_{k_2}^{Thn}} (\beta_j \cdot \overline{w}(u_j) + \gamma_j \cdot \overline{w}(u_j) + \delta_j \cdot \overline{w}(t_j)) . \end{aligned} \tag{5.77}$$

Now, observe that

$$\overline{w}(u_j) > \overline{w}(t_j) \tag{5.78}$$

for all $j \in \{1, 2, ..., k\}$, since $w(u_j) < w(t_j)$. Moreover,

$$\beta_r \cdot \overline{w}(u_r) + \delta_r \cdot \overline{w}(t_r) = \alpha \cdot w(u_r) + \delta_r \cdot \overline{w}(s_r) \tag{5.79}$$

by the definition of β_r. Thus, the expected gain (5.77) is greater than or equal to $Gain({}_CT(n), \sigma_k, c)$ of (5.76).

If Player 2 chooses to start with a vertex t_r in B_r where B_r is a thin branch, then

$$r \in \{i_1^{Thn}, i_2^{Thn}, ..., i_{k_3}^{Thn}\},$$

and the expected gain of Player 1 with the strategy σ_k is

$$\begin{aligned} Gain({}_CT(n),\sigma_k,Z(t_r)) = \alpha\cdot w(u_r) &+ \sum_{j=i_1^{Thk}}^{i_{k_1}^{Thk}} \beta_j\cdot w(u_r) \\ &+ \sum_{j=i_1^{Med}}^{i_{k_2}^{Med}} (\beta_j\cdot w(u_r)+\gamma_j\cdot \overline{w}(u_j)) \\ &+ \beta_r\cdot w(t_r)+\gamma_r\cdot 0+\delta_r\cdot\overline{w}(s_r) \\ &+ \sum_{j=i_1^{Thn},\, j\neq r}^{i_{k_3}^{Thn}} (\beta_j\cdot w(u_r)+\gamma_j\cdot\overline{w}(u_j)+\delta_j\cdot\overline{w}(u_j)) . \end{aligned} \tag{5.80}$$

Now, observe that

$$w(u_r) > \overline{w}(u_j) \tag{5.81}$$

for all $j\in\{1,2,...k\}$. If $j\neq r$, it is clear since the branch at u_r in which lies the centroid includes the edges in the branch at the centroid in which lies u_j. If $j=r$, we have the result by Lemma 5.3. Moreover,

$$\beta_r\cdot w(t_r) = \gamma_r\cdot\overline{w}(t_r) \tag{5.82}$$

by the definition of γ_r. Thus, the expected gain (5.80) is greater than or equal to $Gain({}_CT(n),\sigma_k,u_r)$ of (5.76).

If Player 2 chooses to start with a vertex s_r in B_r where B_r is a thin branch, then

$$r\in\{i_1^{Thn}, i_2^{Thn}, ..., i_{k_3}^{Thn}\},$$

and the expected gain of Player 1 with the strategy σ_k is

$$\begin{aligned} Gain({}_CT(n),\sigma_k,Z(s_r)) = \alpha\cdot w(t_r) &+ \sum_{j=i_1^{Thk}}^{i_{k_1}^{Thk}} \beta_j\cdot w(u_r) \\ &+ \sum_{j=i_1^{Med}}^{i_{k_2}^{Med}} (\beta_j\cdot w(u_r)+\gamma_j\cdot w(u_r)) \\ &+ \beta_r\cdot w(t_r)+\gamma_r\cdot w(s_r)+\delta_r\cdot 0 \\ &+ \sum_{j=i_1^{Thn},\, j\neq r}^{i_{k_3}^{Thn}} (\beta_j\cdot w(u_r)+\gamma_j\cdot w(u_r)+\delta_j\cdot\overline{w}(u_j)) . \end{aligned} \tag{5.83}$$

This expected gain is greater than or equal to $Gain({}_C T(n), \sigma_k, t_r)$ by (5.81) and since

$$\alpha \cdot w(u_r) + \delta_r \cdot \overline{w}(s_r) = \alpha \cdot w(t_r) + \gamma \cdot w(s_r) \tag{5.84}$$

by the definition of δ_r.

If Player 2 chooses to start with a vertex v_j in B_r, where B_r is a thin branch and $v_j \neq u_r, t_r, s_r$, the payoff to Player 1 on all vertices not part of the branch B_r can only increase since Player 2's starting vertex is at a greater distance. Specifically, the payoff to Player 1 on the centroid is now at least $w(u_r)$. Moreover, let us regard the centroid as the root of the tree.

If v_j is a descendant of u_r but not of t_r and s_r then

$$\begin{aligned} Gain({}_C T(n), Z(u_r), Z(v_j)) &\geq w(u_r) \\ Gain({}_C T(n), Z(t_r), Z(v_j)) &\geq \overline{w}(t_r) \\ Gain({}_C T(n), Z(s_r), Z(v_j)) &\geq \overline{w}(s_r). \end{aligned} \tag{5.85}$$

If v_j is a descendant of u_r and t_r but not of s_r then

$$\begin{aligned} Gain({}_C T(n), Z(u_r), Z(v_j)) &\geq w(u_r) \\ Gain({}_C T(n), Z(t_r), Z(v_j)) &\geq w(t_r) \\ Gain({}_C T(n), Z(s_r), Z(v_j)) &\geq \overline{w}(s_r). \end{aligned} \tag{5.86}$$

If v_j is a descendant of u_r, t_r and s_r, then

$$\begin{aligned} Gain({}_C T(n), Z(u_r), Z(v_j)) &\geq w(u_r) \\ Gain({}_C T(n), Z(t_r), Z(v_j)) &\geq w(t_r) \\ Gain({}_C T(n), Z(s_r), Z(v_j)) &\geq w(s_r). \end{aligned} \tag{5.87}$$

In all these cases, since $w(v) > \overline{w}(v)$ for any vertex v other than the centroid by Lemma 5.3, we have

$$\begin{aligned} &\alpha \cdot Gain({}_C T(n), Z(c), Z(v_j)) + \beta_r \cdot Gain({}_C T(n), Z(u_r), Z(v_j)) \\ &+ \gamma_r \cdot Gain({}_C T(n), Z(t_r), Z(v_j)) + \delta_r \cdot Gain({}_C T(n), Z(s_r), Z(v_j)) \\ &\geq \alpha \cdot w(u_r) + \beta_r \cdot w(u_r) + \gamma_r \cdot \overline{w}(t_r) + \delta_r \cdot \overline{w}(s_r) \\ &> \alpha \cdot w(u_r) + \gamma_r \cdot \overline{w}(t_r) + \delta_r \cdot \overline{w}(s_r) \\ &= \beta_r \cdot \overline{w}(u_r) + \gamma_r \cdot \overline{w}(t_r) + \delta_r \cdot \overline{w}(t_r) \end{aligned} \tag{5.88}$$

since $\beta_r \cdot \overline{w}(u_r) + \delta_r \cdot \overline{w}(t_r) = \alpha \cdot w(u_r) + \delta_r \cdot \overline{w}(s_r)$ by (5.79). Now,

$$\begin{aligned} Gain({}_CT(n), Z(c), Z(c)) &= 0, \\ Gain({}_CT(n), Z(u_r), Z(c)) &= \overline{w}(u_r), \\ Gain({}_CT(n), Z(t_r), Z(c)) &= \overline{w}(t_r) \text{ and} \\ Gain({}_CT(n), Z(s_r), Z(c)) &= \overline{w}(t_r), \end{aligned} \tag{5.89}$$

so

$$\begin{aligned} &\beta_r \cdot \overline{w}(u_r) + \gamma_r \cdot \overline{w}(t_r) + \delta_r \cdot \overline{w}(t_r) \\ &= \alpha \cdot Gain({}_CT(n), Z(c), Z(c)) + \beta_r \cdot Gain({}_CT(n), Z(u_r), Z(c)) \\ &+ \gamma_r \cdot Gain({}_CT(n), Z(t_r), Z(c)) + \delta_r \cdot Gain({}_CT(n), Z(s_r), Z(c)). \end{aligned} \tag{5.90}$$

Thus, the expected gain of Player 1 when Player 2 chooses the vertex v_j is greater than or equal to the expected gain of Player 1 when Player 2 chooses the centroid. Similarly, we can show that the expected gain of Player 1 when Player 2 chooses to start with a vertex in B_r, where B_r, $1 \leq r \leq k$, is a medium branch or thick branch is greater than the expected gain of Player 1 when Player 2 chooses to start with the centroid.

If Player 2 chooses to start with a vertex in a branch B_i, $i > k$ instead of the centroid, Player 1's payoff on the vertices in the branches $\{B_1, B_2, ..., B_k\}$ can only increase since Player 2's starting vertex is at a greater distance. Player 1's payoff on the centroid, being zero when Player 2 chooses the centroid, also increases. Thus, the expected gain of Player 1 is again greater.

To sum up, the expected gain of Player 1 with the strategy σ_k is minimal when Player 2 chooses the centroid, thus, it is equal to the guaranteed gain of Player 1 with the strategy σ_k.

□

In the following sections, we will apply the CSS Algorithm 5.21 to different cases of centroidal trees in order to compare the guaranteed gain of the safe strategy of Player 1 from the CSS Algorithm with the safety value of Player 1.

5.2.3 Special Case: Centroidal Trees with Only Thick Branches

Let ${}_CT(n)$ be a centroidal tree with d thick branches at the centroid. If we apply the CSS Algorithm 5.21 to ${}_CT(n)$, we obtain a safe strategy for Player 1, σ_k, where the probabilities on the branches B_i, $1 \leq i \leq k \leq d$, are distributed as

- α on the centroid,
- $\beta_i = \left(\frac{w(u_i)}{\overline{w}(u_i)}\right)\alpha$ on the vertex with the lowest weight in the branch B_i for $1 \leq i \leq k$ and
- 0 on all the other vertices,

where α is such that $\alpha + \sum_{i=1}^{k} \beta_i = 1$. The following lemma gives Player 1's guaranteed gain with the strategy σ_k.

Lemma 5.23. *The guaranteed gain of Player 1 with the strategy σ_k determined by applying the CSS Algorithm 5.21 on ${}_CT(n)$ with only thick branches is*

$$GGain({}_CT(n), \sigma_k) = \frac{\sum_{i=1}^{k} w(u_i)}{1 + \sum_{i=1}^{k} \frac{w(u_i)}{\overline{w}(u_i)}} \tag{5.91}$$

where u_i is the vertex with the lowest weight in the branch B_i for $1 \leq i \leq k$.

Proof. From Theorem 5.22, we know

$$GGain({}_CT(n), \sigma_k) = Gain({}_CT(n), \sigma_k, Z(c)) \tag{5.92}$$

where c is the centroid of ${}_CT(n)$, that is, the guaranteed gain of Player 1 is equal to the expected gain of Player 1 when Player 2 chooses the centroid. Moreover,

$$Gain({}_CT(n), \sigma_k, Z(c)) = \alpha \cdot 0 + \sum_{i=1}^{k} \beta_i \cdot \overline{w}(u_i) \tag{5.93}$$

where u_i is the vertex with the lowest weight in the branch at the centroid B_i, $1 \leq i \leq k$. Replacing the values of β_i in (5.93), we have

$$Gain({}_CT(n), \sigma_k, Z(c)) = \alpha \sum_{i=1}^{k} w(u_i). \tag{5.94}$$

Solving $\alpha + \sum_{i=1}^{k} \beta_i = 1$ for α gives

$$\alpha = \frac{1}{1 + \sum_{i=1}^{k} \frac{w(u_i)}{\overline{w}(u_i)}}. \tag{5.95}$$

Thus,

$$GGain({}_CT(n), \sigma_k) = \frac{\sum_{i=1}^{k} w(u_i)}{1 + \sum_{i=1}^{k} \frac{w(u_i)}{\overline{w}(u_i)}}. \tag{5.96}$$

□

In order to evaluate the proximity of the guaranteed gain obtained with the safe strategy from the CSS Algorithm 5.21 to the safety value of Player 1 on a centroidal tree with only thick branches, let us define an opposing strategy for Player 2. Remember that the maximal gain of Player 1 against an opposing strategy for Player 2 is an upper bound on the safety value.

Definition 5.24. Let the strategy ζ_1 be a mixed strategy on ${}_CT(n)$ with only thick branches where $\zeta_1 = (y_1, y_2, ..., y_n)$ and y_j, the probability of choosing the vertex v_j is

$$y_j = \begin{cases} \alpha = \beta_1 \left(\frac{w(u_1)}{\overline{w}(u_1)} + \sum_{i=2}^{k} \frac{w(u_i)-\overline{w}(u_1)}{\overline{w}(u_i)} \right) \\ \qquad + \sum_{i=2}^{k} \frac{(w(u_1)-w(u_i))(w(u_i)-\overline{w}(u_1))}{\overline{w}(u_i)\overline{w}(u_1)}, & \text{if } v_j = c \\ \beta_1, & \text{if } v_j = u_1 \\ \beta_i = \left(\frac{\overline{w}(u_1)}{\overline{w}(u_i)} \right) \beta_1 + \left(\frac{w(u_1)-w(u_i)}{\overline{w}(u_i)} \right), & \text{if } v_j = u_i, \text{ for } 2 \leq i \leq k \\ 0, & \text{otherwise} \end{cases} \tag{5.97}$$

with α, β_i, $1 \leq i \leq k$ known by solving $\alpha + \sum_{i=1}^{k} \beta_i = 1$. Here c is the centroid, u_i is the vertex with the lowest weight in the branch B_i, $1 \leq i \leq k$, and $\{B_1, B_2, ..., B_k\}$ are the same k branches as in the strategy σ_k. Note that all the probabilities are in terms of β_1.

Let us consider the strategy ζ_1 as an opposing strategy for Player 2 and determine Player 1's maximal gain against ζ_1, i.e. the maximal gain Player 1 can receive over all her possible starting vertices when Player 2 chooses the mixed strategy ζ_1.

Lemma 5.25. *The maximal gain of Player 1 when Player 2 uses the opposing strategy* ζ_1 *on* ${}_CT(n)$ *is*

$$MGain({}_CT(n), \zeta_1) = \overline{w}(u_1) - \frac{\overline{w}(u_1)}{1 + \sum_{i=1}^{k} \frac{w(u_i)}{\overline{w}(u_i)} + \sum_{i=2}^{k} (w(u_1) - w(u_i)) \left(1 + \frac{w(u_i)-\overline{w}(u_1)}{\overline{w}(u_i)\overline{w}(u_1)}\right)}. \tag{5.98}$$

Proof. Recall from Definition 2.10 that the maximal gain of Player 1 against the strategy ζ_1 for Player 2 is

$$MGain({}_CT(n), \zeta_1) = \max_r Gain({}_CT(n), Z(v_r), \zeta_1) \tag{5.99}$$

where $1 \leq r \leq n$. In other words, it is the maximal expected gain Player 1 can get over all her possible starting vertices.

(a) If $v_r = c$, i.e. Player 1 chooses the centroid c as starting vertex, her expected gain is

$$\begin{aligned} Gain({}_CT(n), Z(c), \zeta_1) &= \alpha \cdot Gain({}_CT(n), Z(c), Z(c)) \\ &+ \sum_{i=1}^{k} \beta_i \cdot Gain({}_CT(n), Z(c), Z(u_i)). \end{aligned} \tag{5.100}$$

Since $Gain({}_CT(n), Z(c), Z(c)) = 0$ and $Gain({}_CT(n), Z(c), Z(u_i)) = w(u_i)$, we have

$$Gain({}_CT(n), Z(c), \zeta_1) = \sum_{i=1}^{k} \beta_i w(u_i). \tag{5.101}$$

Replacing the values of β_i, $2 \leq i \leq k$ gives

$$Gain({}_CT(n), Z(c), \zeta_1) = \beta_1 \cdot w(u_1) + \sum_{i=2}^{k} \left(\left(\frac{\overline{w}(u_1)}{\overline{w}(u_i)} \right) \beta_1 + \frac{w(u_1) - w(u_i)}{\overline{w}(u_i)} \right) w(u_i). \tag{5.102}$$

Solving $\alpha + \sum_{i=1}^{k} \beta_i = 1$ for β_1 gives

$$\beta_1 = \frac{1}{1 + \sum_{i=1}^{k} \frac{w(u_i)}{\overline{w}(u_i)} + \sum_{i=2}^{k} (w(u_1) - w(u_i)) \left(1 + \frac{w(u_i) - \overline{w}(u_1)}{\overline{w}(u_i)\overline{w}(u_1)} \right)}. \tag{5.103}$$

If we replace β_1 in (5.102) and simplify, we have

$$\begin{aligned} Gain({}_CT(n), Z(c), \zeta_1) &= \overline{w}(u_1) \\ &- \frac{\overline{w}(u_1)}{1 + \sum_{i=1}^{k} \frac{w(u_i)}{\overline{w}(u_i)} + \sum_{i=2}^{k} (w(u_1) - w(u_i)) \left(1 + \frac{w(u_i) - \overline{w}(u_1)}{\overline{w}(u_i)\overline{w}(u_1)} \right)}. \end{aligned} \tag{5.104}$$

(b) If $v_r = u_i$, i.e. Player 1 chooses a vertex u_i, $1 \leq i \leq k$, her expected gain is

$$\begin{aligned} Gain({}_CT(n), Z(u_i), \zeta_1) &= \alpha \cdot Gain({}_CT(n), Z(u_r), Z(c)) \\ &+ \sum_{i=1}^{k} \beta_i \cdot Gain({}_CT(n), Z(u_r), Z(u_i)). \end{aligned} \tag{5.105}$$

Since $Gain({}_CT(n), Z(u_r), Z(c)) = \overline{w}(u_r)$, $Gain({}_CT(n), Z(u_r), Z(u_i)) = \overline{w}(u_i)$ if $i \neq r$, and $Gain({}_CT(n), Z(u_r), Z(u_i)) = 0$ if $i = r$, we have

$$\begin{aligned} Gain({}_CT(n), Z(u_i), \zeta_1) &= \alpha \cdot \overline{w}(u_r) + \sum_{i=1,\ i\neq r}^{k} \beta_i \cdot \overline{w}(u_r) \\ &= (1 - \beta_r) \cdot \overline{w}(u_r). \end{aligned} \tag{5.106}$$

Replacing the value of β_r gives

$$Gain({}_CT(n), Z(u_i), \zeta_1) = \left(1 - \left(\left(\frac{\overline{w}(u_1)}{\overline{w}(u_r)}\right)\beta_1 + \frac{w(u_1) - w(u_r)}{\overline{w}(u_r)}\right)\right) \cdot \overline{w}(u_r). \tag{5.107}$$

After simplifications, we have

$$Gain({}_CT(n), Z(u_i), \zeta_1) = (1 - \beta_1) \cdot \overline{w}(u_1) \tag{5.108}$$

which is equivalent to

$$\begin{aligned} Gain({}_CT(n), Z(u_i), \zeta_1) &= \overline{w}(u_1) \\ &- \frac{\overline{w}(u_1)}{1 + \sum_{i=1}^{k} \frac{w(u_i)}{\overline{w}(u_i)} + \sum_{i=2}^{k}(w(u_1) - w(u_i))\left(1 + \frac{w(u_i) - \overline{w}(u_1)}{\overline{w}(u_i)\overline{w}(u_1)}\right)} \end{aligned} \tag{5.109}$$

if we replace the value of β_1.

(c) If $v_r \in B_r$ with $r > k$, the expected gain of Player 1 is

$$\begin{aligned} Gain({}_CT(n), Z(v_r), \zeta_1) &= \alpha \cdot Gain({}_CT(n), Z(v_r), Z(c)) \\ &+ \sum_{i=1}^{k} \beta_i \cdot Gain({}_CT(n), Z(v_r), Z(u_i)). \end{aligned} \tag{5.110}$$

Furthermore,

$$Gain({}_CT(n), Z(v_r), Z(c)) \leq \overline{w}(u_r) \text{ and } Gain({}_CT(n), Z(v_r), Z(u_i)) \leq \overline{w}(u_r)$$

where u_r is the vertex with the lowest weight in the branch B_r. Thus,

$$Gain({}_CT(n), Z(v_r), \zeta_1) \leq \alpha \cdot \overline{w}(u_r) + \sum_{i=1}^{k} \beta_i \cdot \overline{w}(u_r) = \overline{w}(u_r). \tag{5.111}$$

This gain is less than the guaranteed gain of Player 1 with the strategy σ_k since otherwise, $\overline{w}(u_r) \geq GGain({}_CT(n), \sigma_k)$ would mean that the branch B_r

satisfies the condition to continue the CSS Algorithm 5.21 to a new strategy σ_{k+1} including some positive probabilities on the branch B_r. Thus, $\overline{w}(u_r) < GGain({}_CT(n), \sigma_k) \leq$ Safety value of Player 1 on ${}_CT(n) \leq MGain({}_CT(n), \zeta_1)$. That is, the gain of Player 1 if she chooses a vertex in a branch other than $\{B_1, B_2, ..., B_k\}$ is not the maximal gain.

(d) If $v_r = t_r$ where t_r is the vertex with the second lowest weight in the branch B_r, $1 \leq r \leq k$, the expected gain of Player 1 is

$$\begin{aligned} Gain({}_CT(n), Z(t_r), \zeta_1) = \alpha \cdot Gain({}_CT(n), Z(t_r), Z(c)) \\ + \sum_{i=1}^{k} \beta_i \cdot Gain({}_CT(n), Z(t_r), Z(u_i)). \end{aligned} \tag{5.112}$$

Since $Gain({}_CT(n), Z(t_r), Z(c)) = \overline{w}(t_r)$, $Gain({}_CT(n), Z(t_r), Z(u_i)) = \overline{w}(u_r)$ if $i \neq r$ and $Gain({}_CT(n), Z(t_r), Z(u_i)) = \overline{w}(t_r)$ if $i = r$,

$$Gain({}_CT(n), Z(t_r), \zeta_1) = \alpha \cdot \overline{w}(t_r) + \beta_r \cdot \overline{w}(t_r) + \sum_{i=1,\ i \neq r}^{k} \beta_i \cdot \overline{w}(u_r). \tag{5.113}$$

This gain is less than $Gain({}_CT(n), Z(u_r), \zeta_1)$ of (5.106) since

$$\alpha \geq \left(\frac{\overline{w}(t_r)}{w(t_r) - w(u_r)} \right) \beta_r. \tag{5.114}$$

(e) If v_r is a vertex in the branch B_r, $v_r \neq u_r, t_r$, the expected gain of Player 1 is

$$\begin{aligned} Gain({}_CT(n), Z(v_r), \zeta_1) = \alpha \cdot Gain({}_CT(n), Z(v_r), Z(c)) \\ + \sum_{i=1}^{k} \beta_i \cdot Gain({}_CT(n), Z(v_r), Z(u_i)). \end{aligned} \tag{5.115}$$

Let us regard the centroid as the root of the tree.

If v_r is a descendant of u_r but not of t_r, then

$$\begin{aligned} Gain({}_CT(n), Z(v_r), Z(c)) \leq \overline{w}(v_r) \leq \overline{w}(t_r) \\ Gain({}_CT(n), Z(v_r), Z(u_r)) \leq \overline{w}(v_r) \leq \overline{w}(t_r) \\ Gain({}_CT(n), Z(v_r), Z(u_i)) \leq \overline{w}(u_r). \end{aligned} \tag{5.116}$$

If v_r is descendant of u_r and t_r, then

$$\begin{aligned} Gain({}_CT(n), Z(v_r), Z(c)) &\leq \overline{w}(t_r) \\ Gain({}_CT(n), Z(v_r), Z(u_r)) &\leq \overline{w}(t_r) \\ Gain({}_CT(n), Z(v_r), Z(u_i)) &\leq \overline{w}(u_r). \end{aligned} \tag{5.117}$$

Thus,

$$\begin{aligned} Gain({}_CT(n), Z(v_r), \zeta_1) &\leq \alpha \cdot \overline{w}(t_r) + \beta_r \cdot \overline{w}(t_r) + \sum_{i=1,\ i\neq r}^{k} \beta_i \cdot \overline{w}(u_r) \\ &= Gain({}_CT(n), Z(t_r), \zeta_1). \end{aligned} \tag{5.118}$$

Therefore, the maximal gain of Player 1 against the strategy ζ_1 is

$$\begin{aligned} MGain({}_CT(n), \zeta_1) &= \overline{w}(u_1) \\ &- \frac{\overline{w}(u_1)}{1 + \sum_{i=1}^{k} \frac{w(u_i)}{\overline{w}(u_i)} + \sum_{i=2}^{k} (w(u_1) - w(u_i)) \left(1 + \frac{w(u_i) - \overline{w}(u_1)}{\overline{w}(u_i)\overline{w}(u_1)}\right)}. \end{aligned} \tag{5.119}$$

□

Theorem 5.26. *In the two-player safe game of Competitive Diffusion on a centroidal tree with only thick branches, ${}_CT(n)$, the safety value of Player 1 is between*

$$\frac{\sum_{i=1}^{k} w(u_i)}{1+\sum_{i=1}^{k} \frac{w(u_i)}{\overline{w}(u_i)}}$$

and

$$\overline{w}(u_1) - \frac{\overline{w}(u_1)}{1+\sum_{i=1}^{k} \frac{w(u_i)}{\overline{w}(u_i)}+\sum_{i=2}^{k}(w(u_1)-w(u_i))\left(1+\frac{w(u_i)-\overline{w}(u_1)}{\overline{w}(u_i)\overline{w}(u_1)}\right)}.$$

Proof. This proof is the same as the proof of Theorem 3.5 having the guaranteed gain from Lemma 5.23 and the maximal gain from Lemma 5.25. □

Note: These bounds on the safety value remain true on trees for which the algorithm stops adding positive probabilities before any medium or thin branches are reached.

Corollary 5.27. *In the two-player safe game of Competitive Diffusion on a centroidal tree, ${}_CT(n)$, for which the strategy σ_k obtained by the CSS Algorithm 5.21 has positive probabilities only on the thick branch B_1, the safety value of Player 1 is*

$$w(u_1) - \frac{w(u_1)^2}{n}.$$

Moreover, the guaranteed gain of the safe strategy for Player 1 obtained by the CSS Algorithm reaches the safety value.

Proof. This result directly follows from evaluating the lower bound and upper bound on the safety value of Theorem 5.26 with $k = 1$. □

Corollary 5.28. *In the two-player safe game of Competitive Diffusion on a centroidal tree, ${}_CT(n)$, with d branches, all of them being thick and $\overline{w}(u_1) = \overline{w}(u_2) = ... = \overline{w}(u_d)$, the safety value of Player 1 is*

$$\frac{d\overline{w}(u_1)w(u_1)}{dw(u_1) + \overline{w}(u_1)}. \tag{5.120}$$

Moreover, the guaranteed gain of the safe strategy for Player 1 obtained by the CSS Algorithm reaches the safety value.

Proof. We know from Lemma 5.17 that when the algorithm adds positive probability on a thick branch B_i, the resulting guaranteed gain does not surpass the criterion of B_i, $\overline{w}(u_i)$. Thus, if $\overline{w}(u_1) = \overline{w}(u_2) = ... = \overline{w}(u_d)$, the criterion of the next branch will always be greater than the current expected gain and so the algorithm will add positive probabilities on all the branches of ${}_CT(n)$. Thus, the corollary directly follows from evaluating the lower bound and upper bound on the safety value of Theorem 5.26 with $\overline{w}(u_1) = \overline{w}(u_2) = ... = \overline{w}(u_d)$ and with $k = d$. □

Note: If ${}_CT(n)$ has k thick branches with $\overline{w}(u_1) = \overline{w}(u_2) = ... = \overline{w}(u_k)$ and the algorithm does not include positive probabilities on the other branches of ${}_CT(n)$, the guaranteed gain with the safe strategy from the CSS Algorithm 5.21 reaches the safety value of

$$\frac{k\overline{w}(u_1)w(u_1)}{kw(u_1) + \overline{w}(u_1)}. \tag{5.121}$$

5.2.4 Special Case: Paths, Spiders and Complete Trees Revisited

In this section, we apply the CSS Algorithm 5.21 to the special cases of Paths, Spiders and Complete Trees in order to evaluate the performance of the algorithm on these centroidal trees. We hope that the algorithm gives a safe strategy for Player 1 with a guaranteed gain at least as good as the guaranteed gains associated to the strategies presented in Chapters 3 and 4.

Complete Trees

Let us start with the complete trees. Let $T(m,h)$ be a complete m-ary tree of height h with $n = \frac{m^{h+1}-1}{m-1}$ vertices. We know by Proposition 4.1 that the root of $T(m,h)$ is the centroid. Thus $T(m,h)$ has m branches at the centroid. The following lemma shows that the branches at the centroid of $T(m,h)$ are thick branches.

Lemma 5.29. *The branches at the centroid of $T(m,h)$, a complete m-ary tree of height h, are thick branches.*

Proof. Being a complete m-ary tree, $T(m,h)$ has m branches at its centroid or correspondingly its root. These branches all have the same number of vertices, $\frac{n-1}{m}$, and all have the same configuration. For them to be thick branches, they need to respect the condition from Definition 5.5,

$$w_2 \geq n - w_1 + \frac{w_1^2}{n} \tag{5.122}$$

where w_2 is the second lowest weight in the branch and w_1 is the lowest weight in the branch. From Lemma 5.8, we know that the lowest weight in a branch at the root of $T(m,h)$ is the weight of the vertex adjacent to the root and the second lowest weight is the weight of a vertex adjacent to the first, so a vertex in level 2. The weight of a vertex adjacent to the root is

$$w_1 = (m-1)\left(\frac{n-1}{m}\right) + 1 = \frac{nm-n+1}{m} \tag{5.123}$$

and the weight of a vertex in level 2 is

$$w_2 = (m-1)\left(\frac{n-1}{m}\right) + 1 + (m-1)\left(\frac{\frac{n-1}{m}-1}{m}\right) + 1 = \frac{nm-n+1}{m} + \frac{nm-n+1}{m^2}. \tag{5.124}$$

Furthermore, w_1 and w_2 satisfy(5.122) for $m \geq 2$. Thus, the branches at the centroid of $T(m,h)$ are thick branches. □

Now that we know that all the branches in $T(m,h)$ are thick branches, we can use the results of Section 5.2.3. Since the branches all have the same number of vertices, Corollary 5.28 tells us that the guaranteed gain with the strategy obtained from the CSS Algorithm 5.21 reaches the safety value of

$$\frac{d\overline{w}(u_1)w(u_1)}{dw(u_1)+\overline{w}(u_1)}. \tag{5.125}$$

Replacing $w(u_1)$ and $d = m$ gives

$$\frac{m\left(n-\left(\frac{nm-n+1}{m}\right)\right)\left(\frac{nm-n+1}{m}\right)}{m\left(\frac{nm-n+1}{m}\right)+\left(n-\left(\frac{nm-n+1}{m}\right)\right)} = \frac{(nm-n+1)(n-1)}{m(nm-n+1)+n-1}. \tag{5.126}$$

Therefore, the CSS Algorithm 5.21 performs ideally on complete trees. The strategy μ_1 of Section 4.2 also had a guaranteed gain of this safety value.

Spiders

Let S be a spider with m legs each having l vertices. Since the legs of the spiders are paths, the branches at the centroid of S are thin branches. Furthermore, the three lowest weights in the branches are:

$$\begin{gathered} w_1 = \left(\frac{m-1}{m}\right)(n-1)+1, \qquad w_2 = \left(\frac{m-1}{m}\right)(n-1)+2, \\ \text{and } w_3 = \left(\frac{m-1}{m}\right)(n-1)+3. \end{gathered} \tag{5.127}$$

From Lemma 5.19, we know that when the CSS Algorithm 5.21 adds positive probabilities on a thin branch B_i, the resulting guaranteed gain does not surpass the criterion of B_i. Furthermore, the branches of S all have the same criterion since their three lowest weights are equal. Thus, on a spider with legs of equal length, the algorithm only stops once it has included positive probabilities on all the branches. Let the safe strategy from the output of the algorithm be denoted by σ_m. The strategy σ_m has probabilities

- α on the centroid,
- $\beta_i = \left(\frac{\overline{w}(t_i)(w(u_i)\overline{w}(s_i)+(w(t_i)-w(s_i))(w(t_i)-w(u_i)))}{\overline{w}(s_i)\overline{w}(u_i)\overline{w}(t_i)+w(s_i)w(t_i)(w(s_i)-w(t_i))}\right)\alpha$ on u_i
- $\gamma_i = \left(\frac{w(t_i)}{\overline{w}(t_i)}\right)\beta_i$ on t_i
- $\delta_i = \left(\frac{w(s_i)}{\overline{w}(s_i)}\right)\gamma_i + \left(\frac{w(t_i)-w(u_i)}{\overline{w}(s_i)}\right)\alpha$ on s_i
- 0 on the other vertices of B_i

for $1 \le i \le m$, where $w(u_i) = w_1$, $w(t_i) = w_2$ and $w(s_i) = w_3$ and where u_i, t_i and s_i are the first three vertices in the leg, B_i, of the spider. From Theorem 5.22, we know that the resulting guaranteed gain is

$$GGain(S, \sigma_m) = Gain(S, \sigma_m, Z(c)) \tag{5.128}$$

where c is the centroid of S and $Z(c)$ is the mixed strategy that chooses the centroid c with probability 1 and all the other vertices with probability 0. Furthermore,

$$Gain(S,\sigma_m,Z(c)) = \alpha\cdot 0 + \sum_{i=1}^{m}\left(\beta_i\cdot(n-w_1)+\gamma_i\cdot(n-w_2)+\delta_i\cdot(n-w_2)\right). \tag{5.129}$$

Now for l fixed, the guaranteed gain with the resulting safe strategy from the CSS algorithm on a spider with legs of equal length should increase with the number of legs m. As the number of legs of the spider increases, there are more vertices near the body on which Player 1 can disperse positive probabilities. In fact, if ${}_m\alpha$ is the probability on the body in the resulting strategy σ_m for a spider with m legs and ${}_{m+1}\alpha$ is the probability on the body in the resulting strategy σ_{m+1} for a spider with $m+1$ legs, we can verify that $(1-{}_{m+1}\alpha) > (1-{}_m\alpha)$. Moreover, when Player 2 chooses the body, the payoff to Player 1 on any of the vertices, other than the body, on which she assigns a positive probability is l or $l-1$, no matter the total number of legs in the spider. Thus, as a worst case scenario, we consider spiders which have only three legs. In this case, (5.129) becomes

$$Gain(S,\sigma_3,Z(c)) = \alpha\cdot 0 + 3\left(\beta\cdot(n-w_1)+\gamma\cdot(n-w_2)+\delta\cdot(n-w_2)\right) \tag{5.130}$$

with α, β, γ and δ known from solving $\alpha+3(\beta+\gamma+\delta)=1$. Here are a couple of numerical examples.

Example 5.30. In a spider S with 3 legs each having 50 vertices, the total number of vertices, n, is 151 and the three lowest weights of the branches are $w_1 = 101$, $w_2 = 102$ and $w_3 = 103$. Solving $\alpha+3(\beta+\gamma+\delta)=1$ and replacing in (5.129) gives a guaranteed gain of 48.17. Moreover, we know from Lemma 3.21 that the safety value of Player 1 is less than or equal to 50. Thus, we have a maximum difference of 1.83 between the guaranteed gain with the safe strategy from the CSS Algorithm and the optimal guaranteed gain. This difference is 1.21% of n.

Example 5.31. In a spider S with 3 legs each having 1000 vertices, the total number of vertices, n, is 3001 and the three lowest weights of the branches are $w_1 = 2001$, $w_2 = 2002$ and $w_3 = 2003$. Solving $\alpha+3(\beta+\gamma+\delta)=1$ and replacing in (5.129) gives a guaranteed gain of 976.094. Moreover, we know from Lemma 3.21 that the safety value of Player 1 is less than or equal to 1000. Thus, we have a maximum difference of 23.9 between the guaranteed gain with the safe strategy from the CSS Algorithm and the optimal guaranteed gain. This difference 0.79% of n.

We can see from the examples that as n grows, the guaranteed gain seems to approach the upper bound on the safety value, l. On that account, replacing w_1, w_2 and w_3 of (5.127) and the solved values for α, β, γ and δ in (5.130), we have

$$Gain(S, \sigma_3, Z(c)) = \frac{14n}{43} + \mathcal{O}(1)\ (n \to \infty) \tag{5.131}$$

for n large. The upper bound on the safety value, on the other hand, is $l = \frac{n}{3} + \mathcal{O}(1)$. That gives a difference of $\frac{1}{129}n + \mathcal{O}(1)$ where $\frac{1}{129} = 0.00775$. For $n > 1000$, this difference is greater than the one obtained with the strategy $C_{S2}(k)$ from Theorem 3.24, which was $\sqrt{\frac{l}{m(m-1)}} + \mathcal{O}(1) = \sqrt{\frac{n}{18}} + \mathcal{O}(1)\ (n \to \infty)$. However, a difference of $0.00775n + \mathcal{O}(1)\ (n \to \infty)$ is not unreasonable and the difference should only become better with more legs in the spider. Moreover, we suspect that a smaller difference would be achieved by generalizing the strategy from the CSS Algorithm to include probabilities on more vertices in the branches, since the legs of the spider are paths.

Paths

Let P_n be a path with n vertices, n odd, for P_n to be a centroidal tree. In a path, both branches at the centroid have $\frac{n-1}{2}$ vertices and the three lowest weights in the branches are

$$w_1 = \left(\frac{n-1}{2}\right) + 1,\ w_2 = \left(\frac{n-1}{2}\right) + 2,\ \text{and } w_3 = \left(\frac{n-1}{2}\right) + 3. \tag{5.132}$$

Similarly as with the spiders, the CSS Algorithm 5.21 will include positive probabilities on both branches of the path and the resulting guaranteed gain will be

$$Gain(S, \sigma_2, Z(c)) = \alpha \cdot 0 + 2\left(\beta \cdot (n - w_1) + \gamma \cdot (n - w_2) + \delta \cdot (n - w_2)\right) \tag{5.133}$$

with α, β, γ and δ known from solving $\alpha + 2(\beta + \gamma + \delta) = 1$. Replacing the values of w_1, w_2, w_3, α, β, γ and δ gives

$$Gain(S, \sigma_2, Z(c)) = \frac{3n}{7} + \mathcal{O}(1)\ (n \to \infty) \tag{5.134}$$

for n large. On the other hand, the upper bound on the safety value from Theorem 3.5 is $\frac{n}{2} - \frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1)$ $(n \to \infty)$. That gives a difference of $\frac{n}{14} - \frac{\sqrt{n}}{\sqrt{3}} + \mathcal{O}(1)$. For $n > 100$, this difference is greater than the one obtained with the strategy $C_{S1}(k)$ from Theorem 3.5 which was $\sqrt{n}\left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{3}}\right) + \mathcal{O}(1)$ $(n \to \infty)$. However, again we suspect that a smaller difference would be achieved by generalizing the strategy from the algorithm to include probabilities on more vertices.

Analysing the performance of the algorithm on paths, spiders and complete trees reinforces our initial suspicion of its possible excellent performance on thick branches but weaker performance on path branches. The following section will assess the performance of the algorithm on random trees which have diverse proportions of thick, medium and thin branches.

5.2.5 Centroidal Safe Strategy Algorithm Performance Assessment on Trees in General

In this section, we apply the CSS Algorithm 5.21 to a variety of trees to appraise the guaranteed gain of the safe strategy from the output of the algorithm. To do so, we will need an upper bound on the safety value and we will evaluate the proximity of the guaranteed gain to this upper bound. Recall that the maximal gain of Player 1 against any mixed strategy of Player 2 is an upper bound on the safety value. Thus, we will suggest an opposing strategy for Player 2. However, we make this suggestion unscrupulously since in the end we have no real interest in the strategy of Player 2. Hence, a simple strategy is satisfactory as long as it allows us to evaluate the reliability of the guaranteed gain with the safe strategy of the CSS Algorithm 5.21.

Centroidal Opposing Strategy Algorithm

For the strategy of Player 2 we also suggest a stepwise algorithm, **Centroidal Opposing Strategy (COS) Algorithm**, which includes two major steps. First, consider that Player 2 chooses to start with the centroid. In this case, the maximal gain of Player 1 is the number of vertices in the largest branch at the centroid and it is achieved when Player 1 chooses the corresponding vertex adjacent to the centroid. To reduce this payoff to Player 1, Player 2 can disperse some positive probabilities on the first vertices of the largest branches. Hence, the first step of the algorithm consists of determining these branches and the opposing strategy of Player 2 which gives the same expected gain to Player 1 no matter if she chooses the centroid or the first vertex in one of these branches. After this step, the maximal gain of Player 1 against the suggested opposing strategy for Player 2 is achieved on either one of the vertices on which Player 2 assigns a positive probability or on a vertex with the second lowest weight in one of the largest branches. Hence, the second step of the algorithm consists of determining the expected gain of Player 1 on the second vertices in the branches. The ones which give a higher payoff to Player 1 are added in the distribution of the positive probabilities in the opposing strategy for Player 2. In the end, the maximal gain of Player 1 is the maximum amongst her expected gains when she chooses a vertex on which Player 2 assigns a positive probability and when she chooses a vertex on which Player 2 assigns a probability of zero. These steps are summarized in the heuristic algorithm defined below.

Algorithm 5.32. Centroidal Opposing Strategy (COS) Algorithm

INPUT: Centroidal tree with d branches at the centroid for which all the weights of the vertices are known, ${}_CT(n)$.

STEP 1: Include some positive probabilities in the opposing strategy of Player 2 on some vertices adjacent to the centroid.

(a) Order the branches $\{B_1, B_2, ..., B_d\}$ such that the number of vertices in the branch B_i is greater than the number of vertices in the branch B_{i+1}. (Or equivalently, $\overline{w}(u_i) > \overline{w}(u_{i+1})$ where u_j is vertex with the lowest weight in the branch B_j).

(b) Start with $i = 1$ and let the opposing strategy of Player 2, $\xi_{1,0}$, be choosing the centroid with probability α and the vertex u_1 with probability β_1. Solve $Gain({}_CT(n), Z(c), \xi_{1,0}) = Gain({}_CT(n), Z(u_1), \xi_{1,0})$ for α and β_1, knowing $\alpha + \beta_1 = 1$.

(c) While $\overline{w}(u_{i+1}) > Gain({}_CT(n), Z(c), \xi_{1,0})$, increase i to $i+1$ and let the opposing strategy of Player 2, $\xi_{i+1,0}$ be choosing the centroid with probability α and the vertex u_j with probability β_j for $1 \leq j \leq i+1$ where u_j is the vertex with the lowest weight in the branch B_j. Solve

$$\begin{aligned} Gain({}_CT(n), Z(c), \xi_{i+1,0}) &= Gain({}_CT(n), Z(u_1), \xi_{i+1,0}) \\ &= Gain({}_CT(n), Z(u_2), \xi_{i+1,0}) \\ &= ... \\ &= Gain({}_CT(n), Z(u_{i+1}), \xi_{i+1,0}) \end{aligned}$$

for α, β_1, ... , β_{i+1}, knowing $\alpha + \sum_{j=1}^{i+1} \beta_j = 1$.

STEP 2: Include some positive probabilities in the opposing strategy of Player 2 on some vertices at distance 2 from the centroid.

(a) Suppose STEP 1 finishes with $i = k$. Calculate the expected gains of Player 1, $Gain({}_CT(n), Z(t_j), \xi_{k,0})$, when she chooses a second vertex with the lowest weight, t_j, in the branches B_j, $1 \leq j \leq k$.

(b) Consider $Gain({}_CT(n), Z(t_r), \xi_{k,0}) = \max\{Gain({}_CT(n), Z(t_j), \xi_{k,0}), 1 \leq j \leq k\}$. If $Gain({}_CT(n), Z(t_r), \xi_{k,0}) > Gain({}_CT(n), Z(c), \xi_{k,0})$, let the opposing strategy of Player 2, $\xi_{k,1}$ be choosing the centroid with probability α, the vertex u_j with probability β_j, $1 \leq j \leq k$ and the vertex t_r with probability γ_r. Solve

$$\begin{aligned} Gain({}_CT(n), Z(c), \xi_{k,1}) &= Gain({}_CT(n), Z(u_1), \xi_{k,1}) \\ &= Gain({}_CT(n), Z(u_2), \xi_{k,1}) \\ &= ... \\ &= Gain({}_CT(n), Z(u_k), \xi_{k,1}) \\ &= Gain({}_CT(n), Z(t_r), \xi_{k,1}) \end{aligned}$$

for α, β_1, ... , β_k, γ_r, knowing $\alpha + \sum_{j=1}^{i+1} \beta_j + \gamma_r = 1$. On the other hand, if $Gain({}_CT(n), Z(t_r), \xi_{k,0}) < Gain({}_CT(n), Z(c), \xi_{k,0})$, skip to STEP 3.

(c) Repeat step (b), calculating the expected gains at each step, until

$$\max\{Gain({}_CT(n), Z(t_j), \xi_{k,l}), 1 \leq j \leq k\} < Gain({}_CT(n), Z(c), \xi_{k,l})$$

where l is the number of times the step (b) is repeated, that is, the number of vertices at distance 2 from the centroid on which Player 2 assigns a positive probability.

Moreover, if at some point, solving for α, β_j, $1 \leq j \leq k$ and γ_i, $1 \leq i \leq l \leq k$ gives negative values, stop adding probabilities and go to STEP 3.

STEP 3: If $l = 0$ or $l > 0$ and all the probabilities in the resulting opposing strategy for Player 2, $\xi_{k,l}$, are positive, calculate the maximal gain of Player 1 against the strategy $\xi_{k,l}$ of Player 2. If $l > 0$ and some negative probabilities exists, calculate the maximal gain of Player 1 against the strategy $\xi_{k,l-1}$ of Player 2.

OUTPUT: Maximal gain of Player 1 against an opposing strategy for Player 2 on ${}_CT(n)$ that can serve as an upper bound on the safety value of Player 1 in the two player game of Competitive Diffusion on ${}_CT(n)$.

A few clarifications on the algorithm are needed. The first one is concerning calculating the various expected gains of Player 1 against the strategy $\xi_{k,l}$ of Player 2. On that account, we have the following formulas.

NOTE: In what follows, u_i and t_i will represent respectively the vertices with the lowest and second lowest weights in the branch B_i.

(i) When Player 1 chooses the centroid:

$$Gain({}_CT(n), Z(c), \xi_{k,l}) = \alpha \cdot 0 + \sum_{i=1}^{k} \beta_i \cdot w(u_i) + \sum_{j=k_1}^{k_l} \gamma_j \cdot w(u_j) \qquad (5.135)$$

where $\{k_1, ..., k_l\} \in \{1, ..., k\}$. When $l = 0$, i.e. when the strategy of Player 2 does not include positive probabilities on vertices at distance 2 from the centroid, the third term is omitted from the calculation of the expected gain.

(ii) When Player 1 chooses a vertex u_q and the strategy $\xi_{k,l}$ of Player 2 has a positive probability of choosing the vertex u_q:

$$\begin{aligned} Gain({}_CT(n), Z(u_q), \xi_{k,l}) = \alpha \cdot \overline{w}(u_q) + \beta_q \cdot 0 + \sum_{i=1, i \neq q}^{k} \beta_i \cdot \overline{w}(u_q) \\ + \gamma_q \cdot w(t_q) + \sum_{j=k_1, j \neq q}^{k_l} \gamma_j \cdot w(u_j) \end{aligned} \tag{5.136}$$

where $\{k_1, ..., k_j\} \in \{1, ..., k\}$ and where γ_q might be zero. When $l = 0$, the forth and fifth terms are omitted from the calculation of the expected gain.

(iii) When Player 1 chooses a vertex t_q and the strategy $\xi_{k,l}$ of Player 2 has a positive probability of choosing the vertex u_q:

$$\begin{aligned} Gain({}_CT(n), Z(t_q), \xi_{k,l}) = \alpha \cdot \overline{w}(t_q) + \beta_q \cdot \overline{w}(t_q) + \sum_{i=1, i \neq q}^{k} \beta_i \cdot \overline{w}(u_q) \\ + \gamma_q \cdot 0 + \sum_{j=k_1, j \neq q}^{k_l} \gamma_j \cdot \overline{w}(u_q) \end{aligned} \tag{5.137}$$

where $k_j \in \{1, ..., k\}$ and where γ_q might be zero.

Secondly, in STEP 2 (c) we verify the resulting opposing strategy for Player 2, $\xi_{k,l}$ for any negative probabilities. The negative probabilities could arise because of the order in which Player 2 forces positive probabilities on vertices, the position of the vertices with respect to the centroid or the payoffs on the vertices. At each step, we are forcing the expected gain of Player 1 to be equal whether she chooses the vertex v or any other vertices on which Player 2 chooses with a positive probability. In certain scenarios, it could become impossible to equal the expected gains without negative contributions. Perhaps, the negative probabilities could have been avoided with a different ordering of vertices added to the opposing strategy of Player 2 or a different distribution of the probabilities on the vertices. However, finding a better ordering or distribution is tedious work and was considered not being worth it. Nevertheless, these occurrences need to be fixed and to do so, we consider the opposing strategy for Player 2 before any negative probabilities arise. Admittedly, this does not give the optimal opposing strategy for Player 2, but again, we do not care for the opposing strategy of Player 2 except to judge the guaranteed gain of Player 1 with the CSS Algorithm.

Lastly, the maximal gain of Player 1 against the strategy $\xi_{k,l}$ needs to be determined. There a few possibilities for the vertex on which Player 1 reaches her maximal gain against the strategy $\xi_{k,l}$ of Player 2. These all need to be checked. The vertex could be:

(i) A vertex which Player 2 chooses with positive probability. The expected gains on all of these vertices are equal by the definition of the strategy $\xi_{k,l}$. Hence, only one needs to be considered, say the centroid. In this case, the expected gain of Player 1 can be determined by (5.135).

(ii) The vertex, u_{l+1} in the branch B_{l+1}, where $\{B_1, B_2, ..., B_d\}$ is the same ordering of the branches as in STEP 1 of the COS Algorithm 5.32 and where B_l is the last branch on which Player 2 has positive probabilities. The expected gain on this vertex can be determined by

$$Gain({}_CT(n), Z(u_{l+1}), \xi_{k,l}) = \alpha \cdot \overline{w}(u_{l+1}) + \sum_{i=1}^{k} \beta_i \cdot \overline{w}(u_{l+1}) + \sum_{j=k_1}^{k_l} \gamma_j \cdot w(u_j) \tag{5.138}$$

where $\{k_1, ..., k_l\} \in \{1, 2, ..., k\}$ are the l branches for which Player 2 chooses a vertex with the second lowest weight with a positive probability. The expected gain of Player 1 when she chooses a vertex u_r, $r > l + 1$ is less than this since $\overline{w}(u_r) < \overline{w}(u_{l+1})$. It is also clear that choosing a vertex at a distance greater than or equal to 2 from the centroid in the branches $\{B_{l+1}, ..., B_d\}$ will give a smaller expected gain to Player 1.

(iii) The vertex t_r in the branch B_r with $r \notin \{k_1, k_2, ..., k_l\}$, that is, the second vertex with the lowest weight in a branch for which Player 2 only chooses the first vertex with a positive probability. In this case the expected gain of Player 1 is

$$\begin{aligned} Gain({}_CT(n), Z(t_r), \xi_{k,l}) = \alpha \cdot \overline{w}(t_r) + \beta_r \cdot \overline{w}(t_r) + \sum_{i=1, i\neq r}^{k} \beta_i \cdot \overline{w}(u_r) \\ + \sum_{j=k_1}^{k_l} \gamma_j \cdot \overline{w}(u_r) \end{aligned} \tag{5.139}$$

where $\{k_1, ..., k_l\} \in \{1, 2, ..., k\}$ are the l branches for which Player 2 chooses a vertex with the second lowest weight with a positive probability. It is clear that choosing vertices other than u_r and t_r in the branch B_r will give smaller expected gains to Player 1.

(iv) A vertex v_r in the branch B_r with $r \in \{k_1, k_2, ..., k_l\}$, that is, a vertex other than u_r and t_r in a branch for which Player 2 chooses the first and second vertices with positive probability. If Player 2 chooses the centroid, the maximal payoff to Player 1 is no more than $\overline{w}(t_r)$. $\overline{w}(t_r)$ is achieved when v_r is adjacent to t_r or if t_r and v_r are both descendants of u_r and $w(t_r) = w(v_r)$. If Player 2 chooses the vertex u_r or t_r, the maximal payoff to Player 1 is no more than $\overline{w}(s_r)$ no matter if s_r is a descend of t_r and u_r or only u_r, where s_r is the third vertex with the lowest weight in the branch B_r. If Player 2 chooses a vertex at distance 1 or 2 from the centroid on another branch, the maximal payoff to Player 1 is no more than $\overline{w}(u_r)$ since the chosen vertex v_r cannot be closer to the centroid than the chosen vertex of Player 2. Thus,

$$\begin{aligned} Gain({}_CT(n), Z(v_r), \xi_{k,l}) \leq \alpha \cdot \overline{w}(t_r) + \beta_r \cdot \overline{w}(s_r) + \sum_{i=1, i\neq r}^{k} \beta_i \cdot \overline{w}(u_r) \\ + \gamma_r \cdot \overline{w}(s_r) + \sum_{j=k_1}^{k_l} \gamma_j \cdot \overline{w}(u_r) \end{aligned} \tag{5.140}$$

Consequently, the maximal gain of Player 1 against the strategy $\xi_{k,l}$ is less than or equal to the maximum of all these cases. This maximum is the value of the output of the algorithm, to be used as an upper bound of the safety value of Player 1 on ${}_CT(n)$.

Now that we have the COS Algorithm 5.32 to determine an upper bound for the safety value of Player 1 on a centroidal tree ${}_CT(n)$, we want to apply it along with the CSS Algorithm 5.21 to evaluate the quality of the guaranteed gain with the suggested safe strategy on various examples.

Generating Examples of Centroidal Trees

In this section we are going to discuss two methods that were used to generate examples of centroidal trees. First, we need a couple of lemmas.

Lemma 5.33. *In a centroidal tree* ${}_CT(n)$ *with* d *branches at the centroid, if the largest branch at the centroid,* B_1*, has* n_1 *vertices, then the total number of vertices in the other branches,* $\{B_2, B_3, ..., B_d\}$ *must be greater than or equal to* n_1*.*

Proof. The reason this needs to be true comes from the fact that the centroid is the vertex with the lowest weight in ${}_CT(n)$. If the largest branch at the centroid has n_1 vertices, this weight is n_1. Consider the vertex u_1 adjacent to the centroid in the branch B_1. By Lemma 5.2, we know that the weight of u_1 is the number of edges in the branch at u_1 in which lies the centroid. Hence, the weight of u_1 is $\sum_{i=2}^{d} n_i + 1$ where n_i is the number of vertices in the branch B_i. Since the weight of u_1 needs to be greater than the weight of the centroid, we have

$$\sum_{i=2}^{d} n_i + 1 > n_1 \Leftrightarrow \sum_{i=2}^{d} n_i \geq n_1. \tag{5.141}$$

□

Lemma 5.34. *Consider a branch at the centroid in a centroidal tree* ${}_CT(n)$*,* B*, with lowest weight* w_1*, second lowest weight* w_2 *and third lowest weight* w_3*. The second lowest weight can be equal to the third lowest weight if and only if*

$$w_2 - w_1 - 1 \geq n - w_2. \tag{5.142}$$

Proof. Let u be the vertex with the lowest weight, t be the vertex with the second lowest weight and s be the third vertex with the lowest weight. First note that if $w(s) = w(t)$, s cannot be a descendant of t and so s and t are both adjacent to u, by Lemma 5.8. (See Figure 5.9).

Let D be the number of vertices in B which are descendants of neither t or s, E be the number of vertices which are descendants of t but not of s and F be the number of vertices which are descendants of s but not of t. By Lemma 5.2, the weights of the vertices s and t are

$$w(s) = w(u) + D + E + 2 \text{ and } w(t) = w(u) + D + F + 2. \tag{5.143}$$

If $w(s) = w(t)$, then we have $E = F$. Now, recall from Lemma 5.4 that $N(u, s)$ is the number of edges in the branches at u other than the one in which lies the centroid and

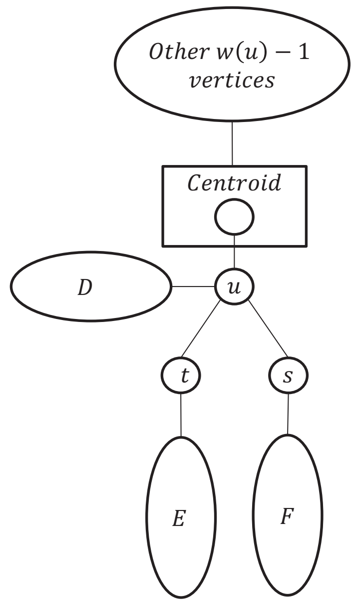


Figure 5.9: Illustration in the proof of Lemma 5.34.

the one in which lies the vertex s. Moreover, $N(u,s) = w(s) - w(u) - 1 = D + E + 1$. Furthermore, $E = F = \overline{w}(t) - 1 = \overline{w}(s) - 1$. Thus,

$$w(s) - w(u) - 1 = D + n - w(s).$$

Since D needs to be greater than or equal to zero, this is equivalent to

$$w(s) - w(u) - 1 \geq n - w(s).$$

□

The centroidal tree examples on which we will apply the CSS Algorithm 5.21 and the COS Algorithm 5.32 were determined randomly by two methods.

Method 1:

- **Fixed parameters**: number of branches at the centroid, d, and number of vertices in the largest branch at the centroid, n_1.
- **Randomized parameters**: number of vertices in the other branches and lowest three weights in the branches.

The randomization in this method is obtained by a function in MATLAB® [19].

`randi([imin,imax],1)`:[20] returns an integer value drawn from the discrete uniform distribution on the interval [imin,imax].

First, the number of vertices in the branches are randomized. The interval of possible values for the number of vertices in the other branches is $[1, n_1]$ with the extra condition on the last branch that the total number of vertices in the branches $B_2, B_3, ..., B_d$ must be greater than or equal to n_1 because of Lemma 5.33. In other words, the number of vertices in the last branch being determined is a random number of the interval $\left[\max\left\{1, n_1 - \sum_{i=2}^{d-1} n_i\right\}, n_1\right]$ where n_i is the number of vertices in the ith branch determined.

After, the lowest three weights in the branches are randomized. In the branch B_i, let u_i be the vertex with the lowest weight, t_i be the vertex with the second lowest weight and s_i be the third vertex with the lowest weight. The weight of u_i is fixed to the value $n - n_i$. The weight of t_i can be any integer in the interval $[n - n_i + 1, n - 1]$. If the weight of t_i is $n - n_i + 1$, then all the other vertices in the branch are descendants of t_i and if the weight of t_i is $n - 1$, then all the other vertices in the branch, including s_i, are leaves of the tree. The possible weights of s_i depend on the weight of t_i. By Lemma 5.34, if $w(t_i) - w(u_i) - 1 \geq \overline{w}(t_i)$, then the weight of s_i can be any integer in the interval $[w(t_i), n - 1]$. Otherwise, the weight of s_i is an integer in the interval $[w(t_i) + 1, n - 1]$.

Method 2:

- **Fixed parameters**: total number of vertices, n.
- **Randomized parameters**: tree structure.

The randomization in this method is obtained by a function of Maple™ in the GraphTheory[RandomGraphs] subpackage [16].

`RandomTree(n)`:[17] creates a random tree on n vertices.

The process of `RandomTree(n)` is stated as follows: "Starting with the empty undirected graph T on n vertices, edges are chosen uniformly at random and inserted into T if they do do not create a cycle. This is repeated until T has $n - 1$ edges." [17]

Once the tree is created, the weights of the vertices and the centroid is determined. Moreover, since we want examples of centroidal trees, if the tree created is bicentroidal, the example is rejected.

With Method 1, we generated 20,000 examples of centroidal trees for each pair of fixed parameters (d, n_1) where $d \in \{2, 3, 5, 10\}$ and $n_1 \in \{100, 1000, 10000\}$. With Method 2, we generated 1,000 examples of centroidal trees for each fixed parameter n, $n \in \{100, 1000\}$. Following this, the CSS Algorithm 5.21 and the COS Algorithm 5.32 were applied on the examples. The programming codes used to generate the examples with Method 1 and Method 2 as well as the programming codes of the CSS and COS algorithms can be found in Appendix A. Afterwards, the difference between the upper bound from the COS algorithm and the guaranteed gain from the CSS algorithm was calculated as a proportion of the weight of the centroid. The reason that the difference was calculated as a proportion of the weight of the centroid and not the total number of vertices, n, is that we know for sure that the safety value of Player 1 is not greater than the weight of the centroid. This is the case since the maximal gain of Player 1 against the mixed strategy of Player 2 when he chooses the centroid with probability 1, is the weight of the centroid. This said, if we want to estimate the loss of payoff by using the safe strategy from the CSS Algorithm instead of the maxmin strategy on a given centroidal tree, the percentage of the weight will give an estimate smaller than the weight of the centroid. On the other hand, if the difference is expressed as a proportion of n, we could have a tree with a large amount of vertices but not necessarily a large centroid weight, for which the estimation of loss could be bigger than the weight of the centroid, an amount that we are sure Player 1 does not reach.

The statistics of the results can be found in Tables 5.1 and 5.2 as well as in Figures 5.10, 5.11, 5.12, 5.13 and 5.14. The figures display two types of graphs: the frequency graph and the cumulative frequency graph. The columns in the frequency graph represent the number of centroidal trees among the 20,000 examples that have a difference between the guaranteed gain with the CSS Algorithm and the upper bound with the COS Algorithm in the intervals $[0, 0], (0, 0.01], (0.01, 0.02], ..., (0.29, 0.30]$ respectively. A column in the cumulative frequency graph represents the number of centroidal trees among the 20,000 examples that have a difference between the guaranteed gain with the CSS Algorithm and the upper bound with the COS Algorithm of less than or equal to the corresponding x-axis value.

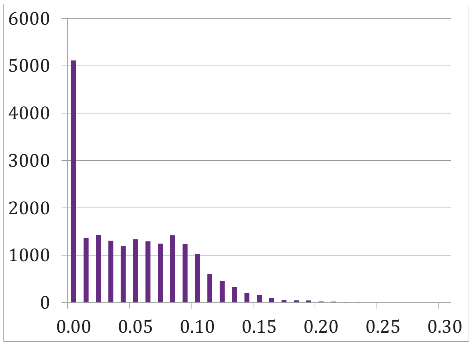
Remarks

Observing the Tables 5.1 and 5.2 as well as the Figures 5.10, 5.11, 5.12, 5.13 and 5.14, we make the following remarks. In the examples obtained randomly by Method 1, as the number of branches at the centroid increases the average difference between the upper bound from the COS Algorithm and the guaranteed gain of the CSS Algorithm decreases. Furthermore, the number of examples for which the CSS Algorithm provides a safe strategy with the optimal guaranteed gain, that is a guaranteed gain equal to the upper bound of the COS Algorithm, increases. Moreover, examples which have a difference between the guaranteed gain with the CSS Algorithm and the upper bound with the COS Algorithm of more than 10% become absent. On the other hand, as the number of vertices in the largest branch increases with the number of branches at the centroid constant, we do not see large variations in the average and maximum differences between the guaranteed gain of the CSS Algorithm and the upper bound of the COS Algorithm. Yet, in most cases, the variation is a slight increase. We suspect that this is due to limiting the positive probabilities on three vertices. It makes sense that this would become more significant as the number of vertices in the largest branch increases. Overall, the CSS Algorithm gives a guaranteed gain nearing the safety value of Player 1 in a large number of the examples from Method 1.

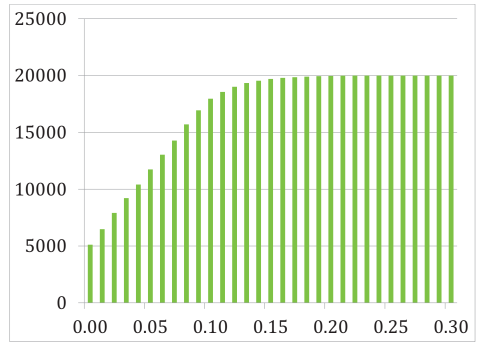
In the examples obtained randomly by Method 2, the average degree of the centroid is 4.23 and 4.40 respectively for $n = 100$ and $n = 1000$ vertices. The average differences between the guaranteed gain with the CSS Algorithm and the upper bound with the COS Algorithm, 0.0841 and 0.1257 are larger than the ones of Method 1. One reason for this could be that in Method 1, the three weights of the branch are determined randomly and so each possible combination of weights should occur with the same probability. However, there are different configurations of tree branches which have the same minimal three weights. While Method 1 provides an analysis over the range of different possibilities, Method 2 can have some combinations of

		d=2	d=3	d=5	d=10
n_1=**100**	Mean:	0.0441	0.0288	0.0143	0.0046
	Standard Deviation:	0.0421	0.0311	0.0203	0.0095
	Maximum Observed:	0.2356	0.2698	0.2506	0.0740
	Number of zeros:	5114	6747	10087	13760
n_1=**1000**	Mean:	0.0455	0.0292	0.0142	0.0045
	Standard Deviation:	0.0440	0.0310	0.0200	0.0093
	Maximum Observed:	0.2410	0.2971	0.2396	0.0826
	Number of zeros:	5115	6520	10042	13674
n_1=**10000**	Mean:	0.0463	0.0296	0.0143	0.0046
	Standard Deviation:	0.0439	0.0316	0.0209	0.0095
	Maximum Observed:	0.2480	0.2874	0.2798	0.0787
	Number of zeros:	4844	6540	10078	13509

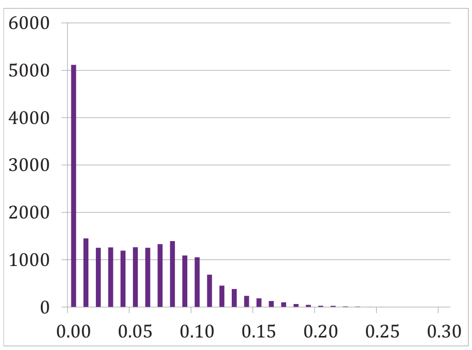
Table 5.1: Statistics on the difference between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 20,000 centroid tree examples obtained by Method 1 for each pair of parameters $d \in \{2, 3, 5, 10\}$ and $n_1 \in \{100, 1000, 10000\}$.



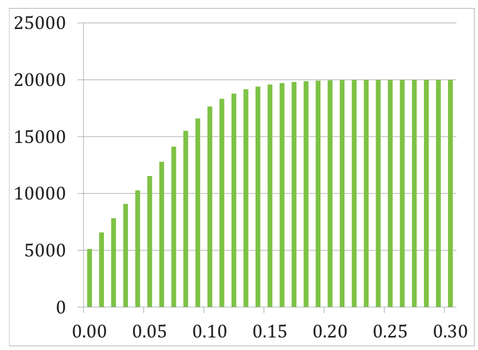
(a) Frequency Graph with $n_1 = 100$.



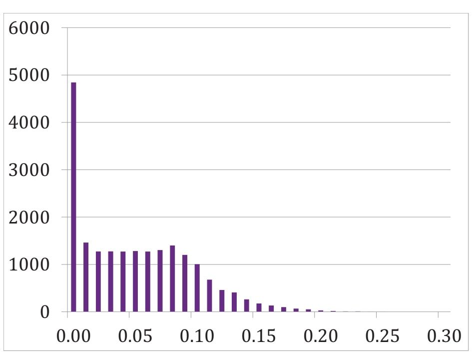
(b) Cumulative Frequency Graph with $n_1 = 100$.



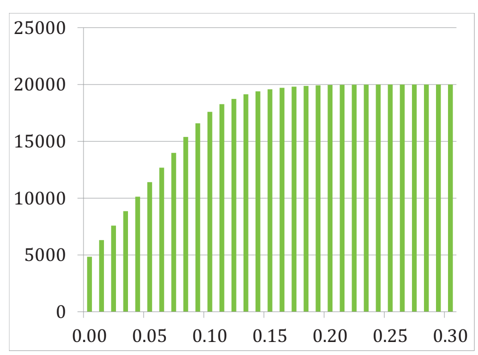
(c) Frequency Graph with $n_1 = 1000$.



(d) Cumulative Frequency Graph with $n_1 = 1000$.

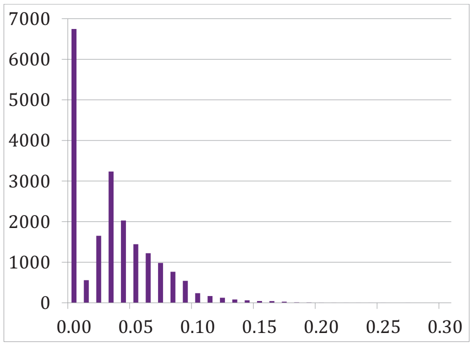


(e) Frequency Graph with $n_1 = 10000$.

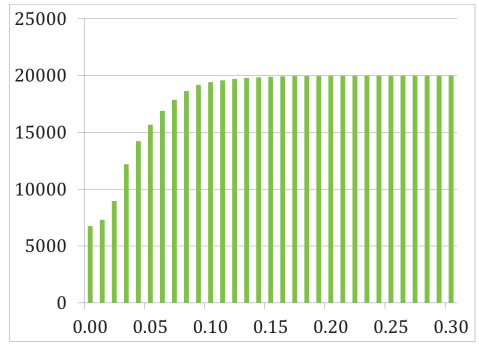


(f) Cumulative Frequency Graph with $n_1 = 10000$.

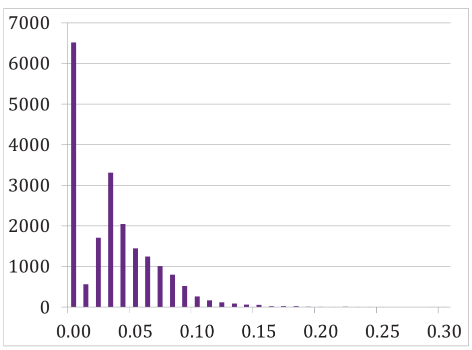
Figure 5.10: Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 20,000 centroidal tree examples obtained by Method 1 with $d = 2$ branches at the centroid and n_1 vertices in the largest branch.



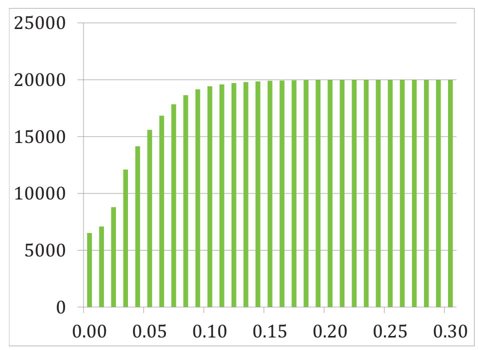
(a) Frequency Graph with $n_1 = 100$.



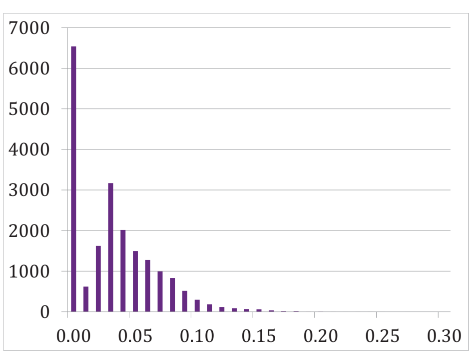
(b) Cumulative Frequency Graph with $n_1 = 100$.



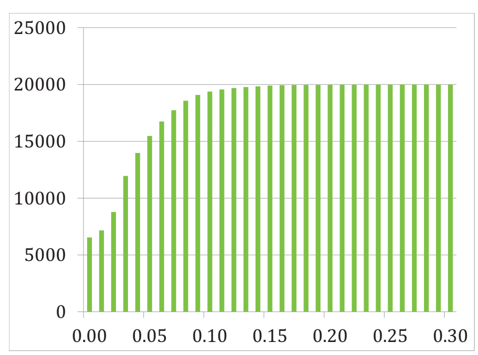
(c) Frequency Graph with $n_1 = 1000$.



(d) Cumulative Frequency Graph with $n_1 = 1000$.

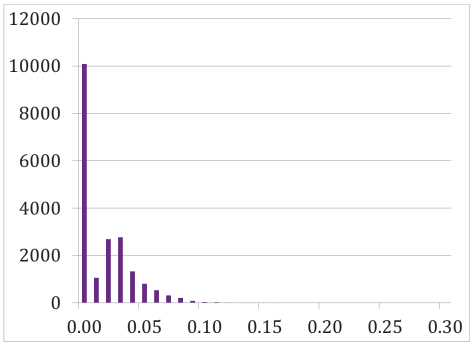


(e) Frequency Graph with $n_1 = 10000$.

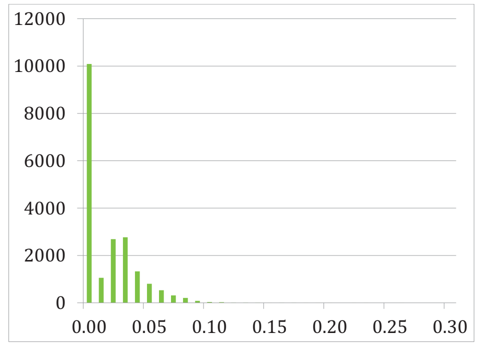


(f) Cumulative Frequency Graph with $n_1 = 10000$.

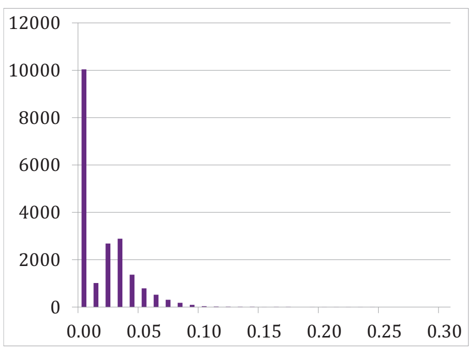
Figure 5.11: Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 20,000 centroidal tree examples obtained by Method 1 with $d = 3$ branches at the centroid and n_1 vertices in the largest branch.



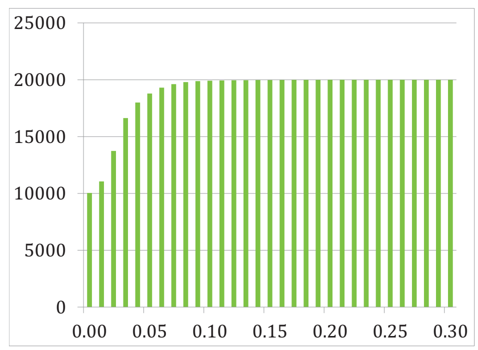
(a) Frequency Graph with $n_1 = 100$.



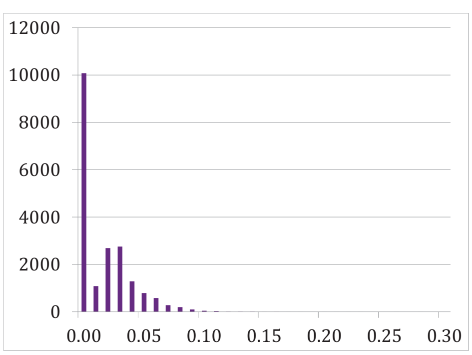
(b) Cumulative Frequency Graph with $n_1 = 100$.



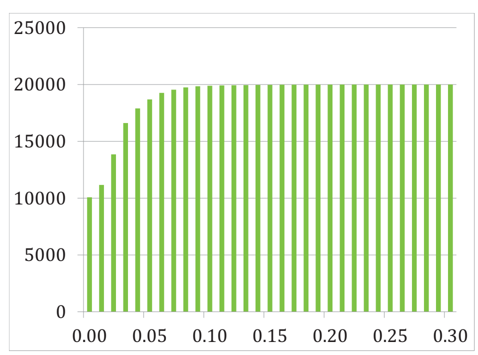
(c) Frequency Graph with $n_1 = 1000$.



(d) Cumulative Frequency Graph with $n_1 = 1000$.

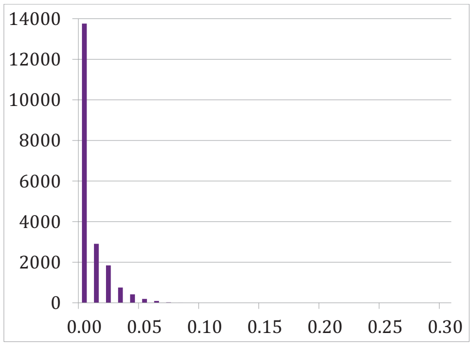


(e) Frequency Graph with $n_1 = 10000$.

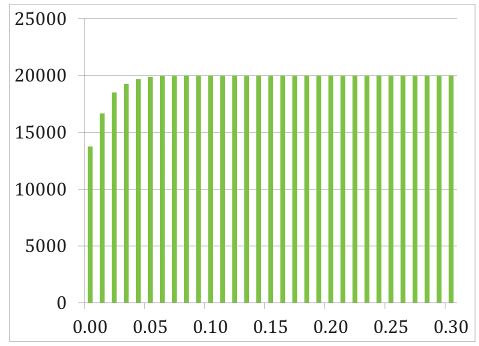


(f) Cumulative Frequency Graph with $n_1 = 10000$.

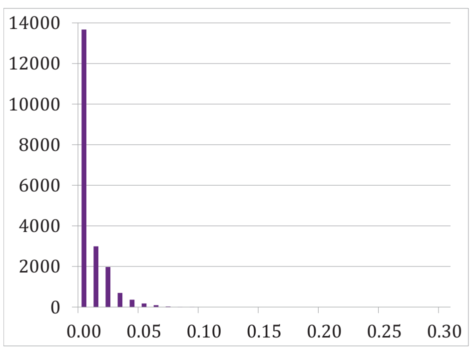
Figure 5.12: Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 20,000 centroidal tree examples obtained by Method 1 with $d = 5$ branches at the centroid and n_1 vertices in the largest branch.



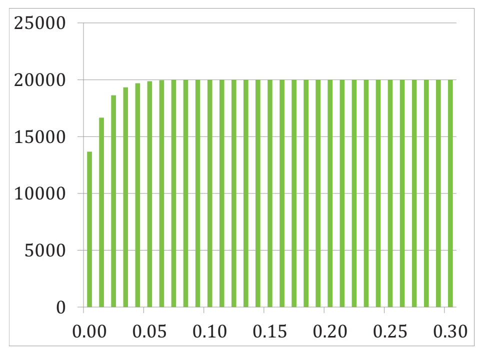
(a) Frequency Graph with $n_1 = 100$.



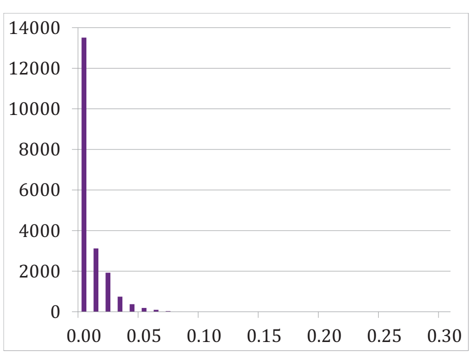
(b) Cumulative Frequency Graph with $n_1 = 100$.



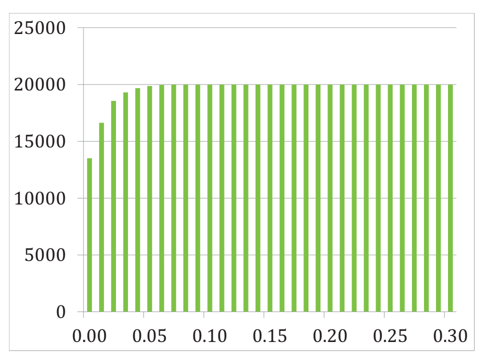
(c) Frequency Graph with $n_1 = 1000$.



(d) Cumulative Frequency Graph with $n_1 = 1000$.



(e) Frequency Graph with $n_1 = 10000$.

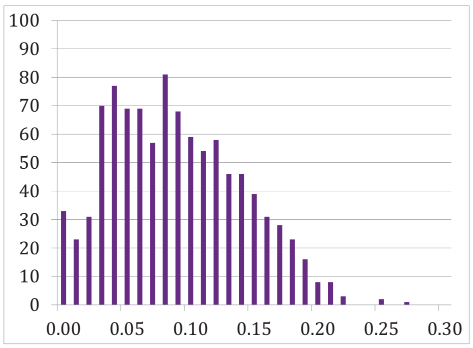


(f) Cumulative Frequency Graph with $n_1 = 10000$.

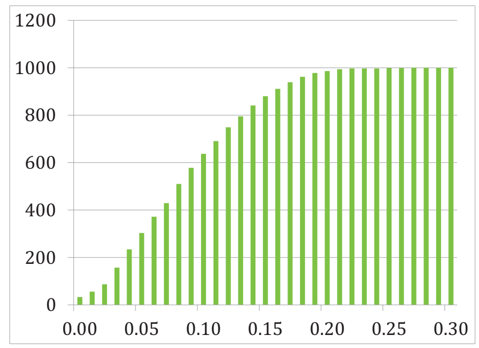
Figure 5.13: Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 20,000 centroidal tree examples obtained by Method 1 with $d = 10$ branches at the centroid and n_1 vertices in the largest branch.

	n=100	**n=1000**
Average d:	4.23	4.40
Minimum d Observed:	3	3
Maximum d Observed:	9	9
Average n_1:	44.19	448.29
Minimum n_1 Observed:	23	280
Maximum n_1 Observed:	49	499
Mean:	0.0841	0.1257
Standard Deviation:	0.0512	0.0526
Maximum Observed:	0.2682	0.2430
Number of zeros:	33	0

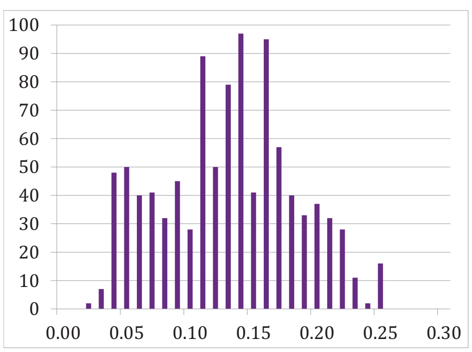
Table 5.2: Statistics on the difference between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a percentage of the weight of the centroid in 1,000 centroid tree examples obtained by Method 2 for $n \in \{100, 1000\}$.



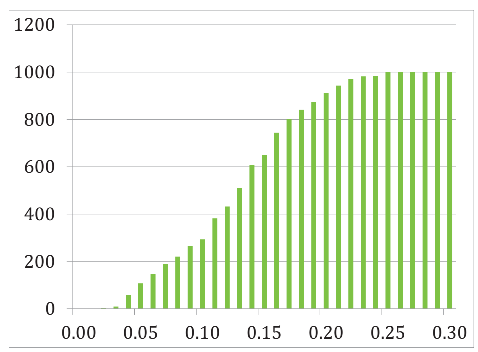
(a) Frequency Graph with $n = 100$.



(b) Cumulative Frequency Graph with $n = 100$.



(c) Frequency Graph with $n = 1000$.



(d) Cumulative Frequency Graph with $n = 1000$.

Figure 5.14: Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm as a proportion of the weight of the centroid in 1,000 centroidal tree examples obtained by Method 2 with the total number of vertices, n.

i	1	2	3
n_i	1000	995	34
$w(u_i)$	1030	1035	1994
$w(t_i)$	1039	1402	2003
$w(s_i)$	1083	1693	2025
Type of branch	Thin	Medium	Thick
Criterion	993.763	741.535	34.000

Table 5.3: Weights in the centroidal tree of Example 5.35.

weights more often than others if they result from a larger number of possible branch configurations. In fact, in the 1,000 examples with $n = 100$, there are 716 examples for which the branch with the largest criterion is a thin branch. With $n = 1000$, this number is 963. With Method 1, on the grand total of 240,000 examples generated, only 13,706 of them had a thin branch as their branch with the largest criterion. That is 5.7% compared to 71.6% and 96.3% with Method 2. Since we have observed that the CSS Algorithm performs better on thick branches than thin branches, the higher average could result from this. Again, we believe that adding positive probabilities on more than three vertices in some thin branches could only help. Nevertheless, if we consider that a difference between the guaranteed gain of the CSS Algorithm and the upper bound from the COS Algorithm of 15% of the weight of the centroid is acceptable, then the CSS Algorithm meets our expectations for 88% of the examples with $n = 100$ and 64.9% of the examples with $n = 1000$. Moreover, one should not forget that this difference can as much be the result of the upper bound with the COS Algorithm being considerably higher than the safety value. Thus, the proximity of the guaranteed gain with the CSS Algorithm to the safety value can only be better.

To finish this section, we present two examples of trees for which the difference between the guaranteed gain with the CSS Algorithm and the upper bound with the COS Algorithm is greater than the majority. The characteristics in the examples are common amongst the trees with higher differences. From them, we hope to suggest possible improvements for the CSS Algorithm 5.21.

Example 5.35. Consider the centroidal tree with 3 branches at the centroid generated by Method 1 with the number of vertices and lowest three weights of Table 5.3. When the CSS Algorithm is applied, positive probabilities are added on branch 1, giving an expected gain of 756.331 to Player 1 when Player 2 chooses to start with the centroid. The algorithm then stops since the criterion of the next branch, 741.535, is smaller than the current expected gain.

In Example 5.35, the algorithm does not add positive probabilities on the second branch since the criterion is smaller than the current expected gain. However, there are 995 vertices in the second branch and this number is greater than the current expected gain. Thus, considered as a thick branch, the criterion of the second branch would be 995 and so adding positive probabilities as described on thick branch would increase the current expected gain. Recall that the thick, medium and thin branches were defined such that the expected gain on the branch when Player 2 chooses the centroid is greater when the positive probabilities are distributed in a certain way. How is it possible then, that a medium branch, for which the expected gain on the branch should be greater when considered as a medium branch, would decrease the current expected gain if added but would increase the current expected gain if added as a thick branch? This is the case because the gain on the different types of branches were calculated when considering the branches isolated, i.e. as the only ones having positive probabilities. Thus, when considering the branches collectively, the splitting between the types of branches might not always be optimal. Nevertheless, it becomes impossible to define branches with numerous possible surroundings. Thus, considering the branches isolated remains, in our eyes, the best approximation. One way to correct this might be to add a step to the algorithm which checks any medium and thin branches with a probability of zero on all their vertices to see if the criterion is greater than the current expected gain when considered as a medium or a thick branch. If so, positive probabilities on these branches with the corresponding distributions could be added. In example 5.35, adding positive probabilities on the second branch increases the guaranteed gain of Player 1 to 803.839 reducing the difference between the bounds of 0.2170 to 0.1901.

To evaluate the amelioration in the guaranteed gain this added step might bring, we included a step which checks any non added medium and thin branches in the examples generated by Method 2 to see if the criterion of the branches when considered as thick branches are greater than the current expected gain. We then added some positive probabilities on the ones that did. The results obtained can be observed in Figure 5.15. The columns in purple and green for each value in the x-axis are the

i	1	2	3	4
n_i	496	366	135	2
$w(u_i)$	504	634	865	998
$w(t_i)$	505	641	866	999
$w(s_i)$	506	687	891	0
Type of branch	Thin	Thin	Medium	Thick
Criterion	495.326	360.041	134.134	0

Table 5.4: Weights in the centroidal tree of Example 5.36.

frequencies without the extra step in the algorithm, i.e. precisely the ones of Figure 5.14. The columns in red or blue, represent the change in the frequencies with the extra step in the algorithm. We see that adding the extra step does reduce the difference in the bounds on some examples. As a group though, the average difference in the bounds only decreases from 0.0841 to 0.0791 with $n = 100$ and from 0.1257 to 0.1237 with $n = 1000$. Still, this is a possible improvement on the CSS Algorithm.

Example 5.36. Consider the centroidal tree with 4 branches at the centroid generated by Method 2, with the number of vertices and the lowest three weights of Table 5.4. The tree is also represented in Figure 5.16.

When the CSS Algorithm is applied, positive probabilities are added on the first branch, B_1, giving an expected gain of 374.881 to Player 1 when Player 2 chooses to start with the centroid. The algorithm then stops since the criterion of the next branch, 360.041, is smaller than the current expected gain.

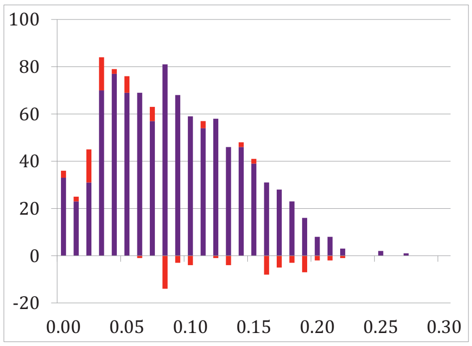
When the COS Algorithm is applied, the opposing strategy for Player 2 from the output, $\xi_{2,2}$ is $\alpha = 0.3182$ on the centroid, $\beta_1 = 0.2801$ on the vertex u_1, $\beta_2 = 0.1228$ on the vertex u_2, $\gamma_1 = 0.2721$ on the vertex t_1, $\gamma_2 = 0.0067$ on the vertex t_2 and 0 on all the other vertices. The expected gains of Player 1 against the strategy $\xi_{2,2}$ are

(i) $Gain({}_CT(n), \xi_{2,2}, Z(c)) = Gain({}_CT(n), \xi_{2,2}, Z(u_1)) = Gain({}_CT(n), \xi_{2,2}, Z(u_2)) = Gain({}_CT(n), \xi_{2,2}, Z(t_1)) = Gain({}_CT(n), \xi_{2,2}, Z(t_2)) = 360.447$

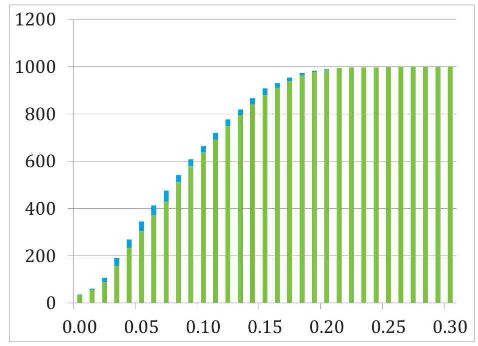
(ii) $Gain({}_CT(n), \xi_{2,2}, Z(u_3)) = 238.762$

(iii) $Gain({}_CT(n), \xi_{2,2}, Z(u_4)) = 142.845$

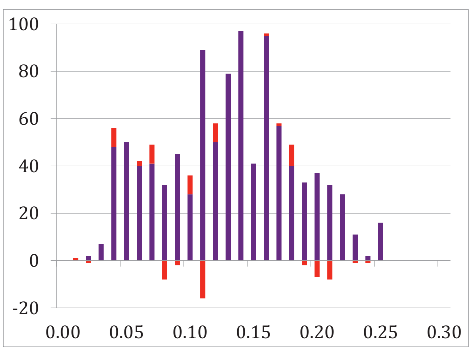
(iv) $Gain({}_CT(n), \xi_{2,2}, Z(v_1)) \leq 494.577$



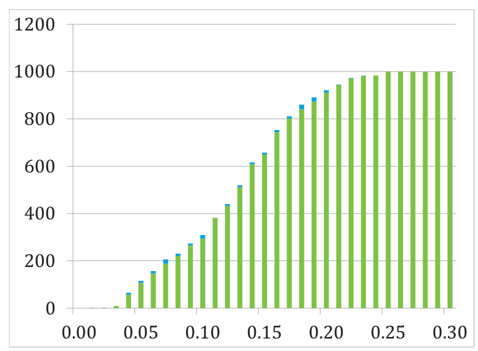
(a) Frequency Graph with $n = 100$.



(b) Cumulative Frequency Graph with $n = 100$.



(c) Frequency Graph with $n = 1000$.



(d) Cumulative Frequency Graph with $n = 1000$.

Figure 5.15: Frequency of the differences between the upper bound of the COS Algorithm and the guaranteed gain of the CSS Algorithm with the added step suggested from Example 5.35 as a proportion of the weight of the centroid in 1,000 centroidal tree examples obtained by Method 2 with the total number of vertices, n.

NOTE: The columns in purple and green are the frequencies with the CSS Algorithm as defined in the Algorithm 5.21, precisely the columns of Figure 5.14. The columns in red and blue represent the change in the frequencies with the CSS Algorithm when the step suggested from Example 5.35 is added.

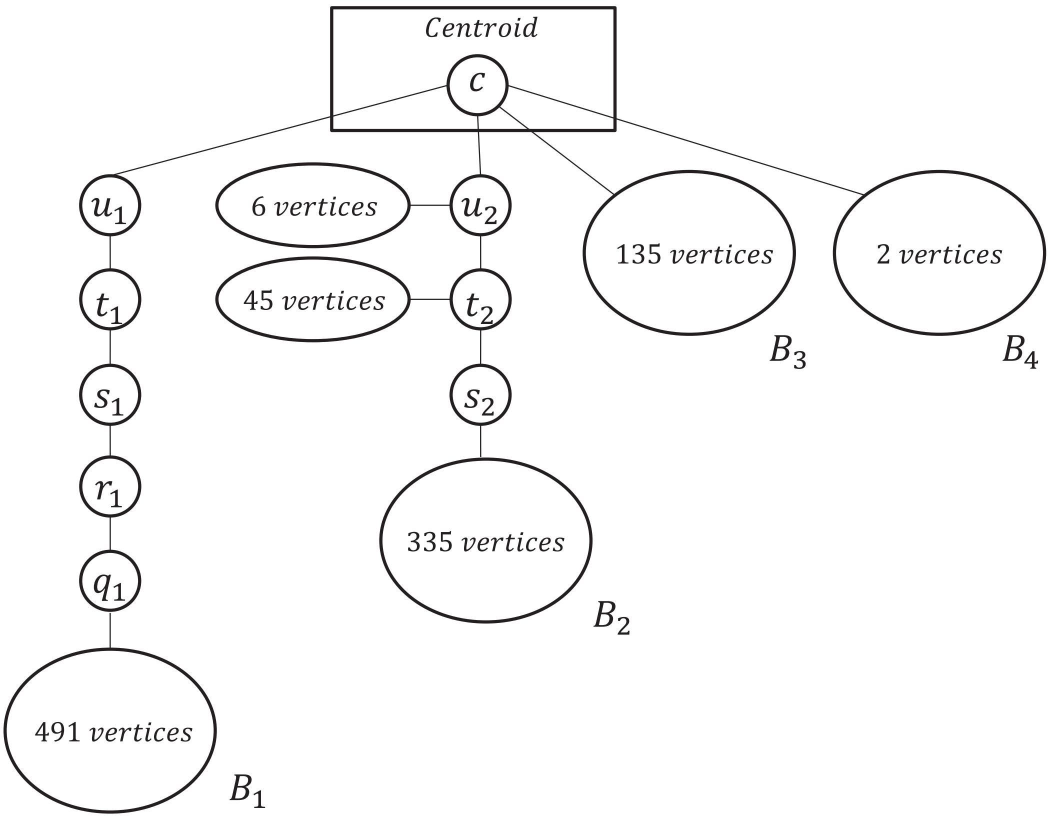


Figure 5.16: Representation of the centroidal tree in Example 5.36.

Centroid		Branch B_i	$i=1$	$i=2$
α	0.0478	β_i	0.0493	0.0515
		γ_i	0.0680	0.0906
		δ_i	0.0947	0.2546
		ϵ_i	0.1688	-
		η_i	0.1747	-

Table 5.5: Mixed strategy for Player 1 on the centroidal tree of Example 5.36.

(v) $Gain({}_CT(n), \xi_{2,2}, Z(v_2)) \leq 356.905$

where v_1 is any vertex other than u_1 and t_1 in the branch B_1 and v_2 is any vertex other than u_2 and t_2 in the branch B_2. The maximum of these values, 494.577 is the upper bound on the safety value output from algorithm. Hence, the difference between the upper bound and the guaranteed gain of Player 1 expressed as a proportion of the weight of the centroid is 0.2415.

In Example 5.36, the first and second branches at the centroid are thin branches. Hence, a higher expected gain might be achievable if Player 1 includes positive probabilities on more vertices of these branches. For instance, consider a mixed strategy for Player 1 where Player 1 incorporates a positive probability of ϵ_1 on the vertex r_1 and η_1 on the vertex q_1 where r_1 and q_1 are the two next vertices in the branch B_1 as shown in Figure 5.16. We can equal the expected gains of Player 1 when Player 2 chooses the centroid, u_1, u_2, t_1, t_2, s_1, s_2, r_1 and q_1 by solving for α, β_1, β_2, γ_1, γ_2, δ_1, δ_2, ϵ_1 and η_1, knowing that $\alpha + \sum_{i=1}^{2}(\beta_i + \gamma_i + \delta_i) + \epsilon_1 + \eta_1 = 1$. This gives the mixed strategy described in Table 5.5 with a guaranteed gain of 417.44 when Player 2 chooses the centroid. This higher expected gain reduces the difference between the guaranteed gain with the CSS Algorithm and the upper bound of the COS Algorithm of 0.2414 to 0.1555.

Alternatively, the upper bound on the safety value might be lowered if Player 2 also includes probabilities on vertices at distance greater than 2 from the centroid on which Player 1 still has a high payoff. Notice that the maximal gain of Player 1 against the opposing strategy $\xi_{2,2}$ of Player 2 is achieved when Player 1 chooses a vertex other than u_1 and t_1 in the branch B_1. Therefore, as much as the CSS Algorithm does not perform at its best on thin branches, we might also be overestimating the maximal gain of Player 1 against an opposing strategy of Player 2. We believe though, that the difference between the guaranteed gain with the CSS Algorithm and the upper bound of the COS Algorithm is more due to the former than the latter case since Player 2 in his opposing strategy, cannot have large positive probabilities on vertices too far from the centroid without giving a high payoff to Player 1 on the centroid.

All in all, the CSS Algorithm 5.21 provides a safe strategy for Player 1 with a reasonable guaranteed gain in a lot of instances. Moreover, it seems a good base on which one could try to incorporate modifications to accommodate the cases on which it is not optimal.

5.3 Bicentroidal Trees

A bicentroidal tree with n vertices, ${}_BT(n)$, has two adjacent vertices as centroid and $n-2$ vertices distributed equally between the two sides of the centroid. The weight of both vertices in the centroid is $\frac{n}{2}$. A reasoning similar to the thick, medium and thin branches of the centroidal trees should be translatable to the bicentroidal trees. However, a few minor modifications might be needed since one centroid vertex is in the branch at the other centroid vertex. Nevertheless, bicentroidal trees always have two branches with $\frac{n}{2}$ vertices, thus an algorithm suggesting a safe strategy for Player 1 should be simpler and so should the analysis of its performance.

Chapter 6

A Few Other Graphs

The chapters 3, 4 and 5 concentrated on the two-player safe game of Competitive Diffusion on trees, i.e. graphs that do not have cycles. One could ask himself what could be a safe strategy for Player 1 in the safe game of Competitive Diffusion on a graph which has one or more cycles. On that account, this chapter gives an introduction to this study by analysing the two-player safe game of Competitive Diffusion on a few other simple families of graphs, precisely cycles, complete graphs and complete bipartite graphs.

6.1 Cycles

Recall from Definition 1.32 that a cycle is a graph that can be drawn so that all its vertices and edges lie on a single circle. We denote a cycle with n vertices by C_n and we label the vertices in the cycle in order, so starting from a vertex v_1 and ending the cycle with the vertex v_n. Since there is no vertex in a circle that is an advantageous choice of starting vertex, we define the following mixed strategy for a player on a cycle C_n in which all the vertices are chosen with equal probability.

Definition 6.1. Let the strategy θ_C be a mixed strategy on C_n where

$$\theta_C = (x_1, x_2, ..., x_n)$$

and x_i, the probability of choosing the vertex v_i, is $\frac{1}{n}$ for all $i \in \{1, 2, ..., n\}$.

Lemma 6.2. *The guaranteed gain of Player 1 with the strategy θ_C on a cycle with n vertices C_n is*

$$GGain(C_n, \theta_C) = \frac{n}{2} + \mathcal{O}(1).\ (n \to \infty) \tag{6.1}$$

Proof. From Definition 2.9, we have that the guaranteed gain of Player 1 with the strategy θ_C is

$$GGain(C_n, \theta_C) = \min_j Gain(C_n, \theta_C, Z(v_j)) \tag{6.2}$$

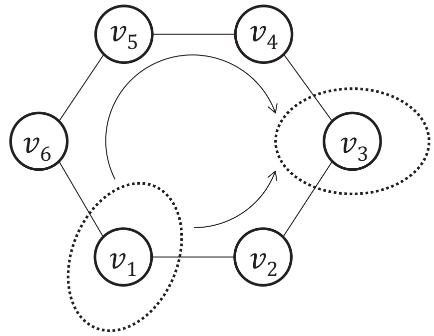


Figure 6.1: Illustration of two paths between the vertices in a cycle.

where $1 \leq j \leq n$ and $Z(v_j)$ is the mixed strategy which chooses the vertex v_j with probability 1. Due to the symmetry of the cycle and the strategy θ_C, the expected gain of Player 1 will be the same no matter the choice of starting vertex for Player 2. Thus, we suppose that Player 2 chooses the vertex v_1. The expected gain of Player 1 with the strategy θ_C when Player 2 chooses to start with the vertex v_1 is

$$Gain(C_n, \theta_C, Z(v_1)) = \frac{1}{n} \sum_{i=1}^{n} Gain(C_n, Z(v_i), Z(v_1)) \tag{6.3}$$

Note that between any two vertices in a cycle there are two paths (see for example Figure 6.1). When Player 1 chooses the vertex v_i, $i \neq 1$, the two paths are:

- A path with i vertices: $v_1 - v_2 - ... - v_i$.
- A path with $n - i + 2$ vertices: $v_1 - v_n - v_{n-1} - ... - v_i$.

We know the payoff of Player 1 on a path from the game matrix in Theorem 3.1. Thus,

$$Gain(C_n, Z(v_i), Z(v_1)) = \pi_{i,1}\big|_i + \pi_{n-i+2,1}\big|_{n-i+2} - 1 \tag{6.4}$$

where $\pi_{i,j}\big|_n$ is the value in the game matrix of a path with n vertices when Player 1 chooses the vertex v_i and Player 2 chooses the vertex v_j as defined in Theorem 3.1 and where we remove 1 since the vertex v_i is counted as a payoff to Player 1 in both paths. Replacing the expressions for $\pi_{i,j}\big|_n$ gives

$$Gain(C_n, Z(v_i), Z(v_1)) = i - \left\lfloor \frac{i+1}{2} \right\rfloor + n - i + 2 - \left\lfloor \frac{n-i+3}{2} \right\rfloor - 1. \tag{6.5}$$

Evaluating for the different parities of n and i, we can simplify to

$$Gain(C_n, Z(v_i), Z(v_1)) = \begin{cases} \frac{n}{2}, & \text{if } n \text{ is even and } i \text{ is even} \\ \frac{n-1}{2}, & \text{if } n \text{ is odd and } i \text{ is even} \\ \frac{n-1}{2}, & \text{if } n \text{ is odd and } i \text{ is odd} \\ \frac{n}{2} - 1, & \text{if } n \text{ is even and } i \text{ is odd.} \end{cases} \tag{6.6}$$

Now, if n is even, there are $\frac{n}{2}$ even i's in $\{2, 3, ..., n\}$ and $\frac{n}{2} - 1$ odd i's in $\{2, 3, ..., n\}$, whereas if n is odd, there are $\frac{n-1}{2}$ even i's in $\{2, 3, ..., n\}$ and $\frac{n-1}{2} + 1$ odd i's in $\{2, 3, ..., n\}$. Therefore, if n is even,

$$Gain(C_n, \theta_C, Z(v_1)) = \frac{1}{n}\left(0 + \left(\frac{n}{2}\right)\left(\frac{n}{2}\right) + \left(\frac{n}{2} - 1\right)\left(\frac{n}{2} - 1\right)\right) \tag{6.7}$$

which simplifies to

$$Gain(C_n, \theta_C, Z(v_1)) = \frac{n}{2} - 1 + \frac{1}{n} = \frac{n}{2} + \mathcal{O}(1) \tag{6.8}$$

Similarly, if n is odd,

$$Gain(C_n, \theta_C, Z(v_1)) = \frac{n}{2} - \frac{1}{2} = \frac{n}{2} + \mathcal{O}(1). \tag{6.9}$$

□

Lemma 6.3. *The maximal gain of Player 1 on a cycle with n vertices, C_n, when Player 2 chooses any vertex v_j for $j \in \{1, 2, ...n\}$ is*

$$MGain(C_n, Z(v_j)) = \frac{n}{2} + \mathcal{O}(1) \ (n \to \infty). \tag{6.10}$$

Proof. Recall from Definition 2.10 that the maximal gain of Player 1 when Player 2 chooses the vertex v_j is

$$MGain(C_n, Z(v_j)) = \max_i Gain(C_n, Z(v_i), Z(v_j)) \tag{6.11}$$

where $1 \leq i \leq n$. We can always relabel the vertices in the cycle if necessary and suppose that the chosen vertex by Player 2, v_j, is the vertex v_1. If Player 1 also chooses the vertex v_1, her gain is zero. On the other hand, if Player 1 chooses a vertex v_i, $i \in \{2, ..., n\}$, we know from (6.6) that

$$Gain(C_n, Z(v_i), Z(v_1)) = \frac{n}{2} + \mathcal{O}(1). \tag{6.12}$$

Since this is true for any vertex v_i, $i \in \{2, ..., n\}$, this is the maximal gain of Player 1 on C_n. □

Theorem 6.4. *In the two-player Competitive Diffusion game on C_n, the safety value of Player 1 is $\frac{n}{2} + \mathcal{O}(1)$ $(n \to \infty)$.*

Proof. This proof is essentially the same as the proof of Theorem 3.5 having the guaranteed gain from Lemma 6.2 and the maximal gain from Lemma 6.3 except that in this case, the bounds are equal. Thus, the safety value of Player 1 is exactly $\frac{n}{2} + \mathcal{O}(1)$ □

Let us compare the safety value with the payoffs in the pure Nash equilibrium described in Theorem 2.6.

Theorem 6.5. *For a cycle with n vertices C_n, the safety value of Player 1 is equal to the gain of any player in a Nash equilibrium $\frac{n}{2} + \mathcal{O}(1)$ $(n \to \infty)$.*

Proof. This result directly comes from comparing the safety value in Theorem 6.4 to the payoffs in the Nash equilibrium of Theorem 2.6. □

6.2 Complete Graphs

Recall from Definition 1.41 that a complete graph is a graph in which every pair of vertices is joined by an edge. We denote a complete graph on n vertices by K_n. Since every edge in the graph is present, there is no vertex in the graph that is an advantageous choice of starting vertex. Hence, we define the following strategy for a player on a complete graph K_n in which all the vertices are chosen with equal probability.

Definition 6.6. Let the strategy θ_K be a mixed strategy on K_n where

$$\theta_K = (x_1, x_2, ..., x_n)$$

and x_i, the probability of choosing the vertex v_i, is $\frac{1}{n}$ for all $i \in \{1, 2, ..., n\}$.

Lemma 6.7. *The guaranteed gain of Player 1 with the strategy θ_K on a complete graph with n vertices, K_n is*

$$GGain(K_n, \theta_K) = 1 - \frac{1}{n}. \tag{6.13}$$

Proof. From Definition 2.9, we know that the guaranteed gain of Player 1 with the strategy θ_K is

$$GGain(K_n, \theta_K) = \min_j Gain(K_n, \theta_K, Z(v_j)) \tag{6.14}$$

where $1 \leq j \leq n$. Due to the symmetry in the complete graph and the strategy θ_K, the expected gain of Player 1 will be the same no matter the choice of starting vertex for Player 2. Thus, we suppose that Player 2 chooses the vertex v_1. The expected gain of Player 1 with the strategy θ_K when Player 2 chooses to start with the vertex v_1 is

$$Gain(K_n, \theta_K, Z(v_1)) = \frac{1}{n} \sum_{i=1}^{n} Gain(K_n, Z(v_i), Z(v_1)). \tag{6.15}$$

If Player 1 chooses the vertex v_1, her gain is zero whereas if Player 1 chooses a vertex v_i with $i \neq 1$, her gain is 1 since all the other vertices are neighbours of both v_i and v_1, all the edges in K_n being present. Thus,

$$Gain(K_n, \theta_K, Z(v_1)) = \frac{1}{n} \left(0 + (n-1) \cdot 1\right). \tag{6.16}$$

□

Lemma 6.8. *The maximal gain of Player 1 on a complete graph with n vertices, K_n when Player 2 chooses the strategy θ_K is*

$$MGain(K_n, \theta_K) = 1 - \frac{1}{n} \tag{6.17}$$

Proof. Recall from Definition 2.10 that the maximal gain of Player 1 when Player 2 chooses the strategy θ_K is

$$MGain(K_n, \theta_K) = \max_i Gain(K_n, Z(v_i), \theta_K) \tag{6.18}$$

where $1 \leq i \leq n$. Due to the symmetry in the complete graph and the strategy θ_K, the expected gain of Player 1 will be the same no matter her choice of starting vertex. Suppose Player 1 chooses the vertex v_1. The expected gain of Player 1 is

$$Gain(K_n, Z(v_1), \theta_K) = \frac{1}{n} \sum_{j=1}^{n} Gain(K_n, Z(v_1), Z(v_j)). \tag{6.19}$$

When Player 2 also chooses the vertex v_1, the payoff to Player 1 is zero. On the other hand, if Player 2 chooses a vertex v_j, $j \neq 1$, the payoff to Player 1 is 1 since all the other vertices are neighbours of both v_j and v_1, all the edges in K_n being present. Thus,

$$Gain(K_n, Z(v_1), \theta_K) = \frac{1}{n}(0 + (n-1) \cdot 1). \tag{6.20}$$

□

Theorem 6.9. *In the two-player Competitive Diffusion game on K_n, the safety value of Player 1 is $1 - \frac{1}{n}$.*

Proof. This proof is similar to the proof of Theorem 3.5 having the guaranteed gain from Lemma 6.7 and the maximal gain from Lemma 6.8 of equal value $1 - \frac{1}{n}$. □

The complete graphs are not interesting graphs for the two-player Competitive Diffusion since the gain of both players is always either 0 or 1. However, it informs us that the number of vertices part of a cluster in a large graph gained by Player 1 would be negligible when the players choose starting vertices that are at equal distance from the cluster. For example, consider the portion of a graph representing a social network in Figure 6.2. If the players choose the starting vertices circled in Blue and Yellow, they are at equal distance from the cluster circled in Red. Thus, after two waves of diffusion, the colors of both players will reach a vertex of the cluster. In the following wave, most of the vertices in the cluster will turn grey since they are neighbours of vertices in both colors.

6.3 Complete Bipartite Graphs

Recall from Definition 1.45 that a complete bipartite graph is a graph whose vertices can be partitioned into two subsets U and W such that every vertex in U is joined to every vertex in W. We denote a complete bipartite graph that has m vertices in the subset U and n vertices in the subset W by $K_{m,n}$ and we label the vertices in the subset U by $\{u_1, u_2, ..., u_m\}$ and the vertices in the subset W by $\{w_1, w_2, ..., w_n\}$. Let us consider strategies which have a probability of α on the vertices of the subset U and β on the vertices of the subset W. Following a similar idea as the strategy $C_{S2}(k)$ on the spiders and the strategies μ_1 and μ_2 on the complete trees, we assume such a strategy for Player 1 and we determine Player 1's expected gain when Player 2 chooses a vertex in the subset U and when Player 2 chooses a vertex in the subset W.

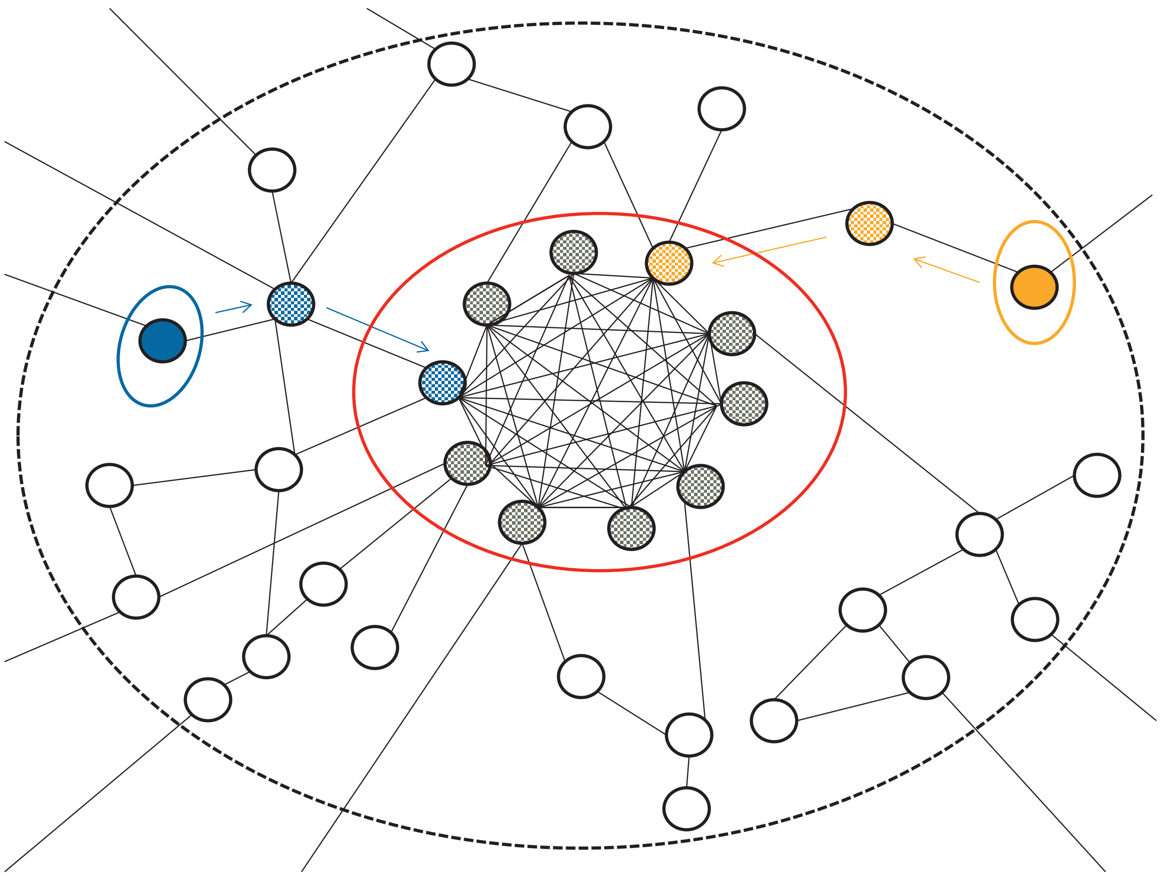


Figure 6.2: Example of two players choosing starting vertices at equal distance from a cluster in a large graph.

Then, we force the two expected gains to be equal by solving for α and β knowing that $m\alpha + n\beta = 1$. Similarly, we can assume that Player 2 has a strategy where he chooses a vertex from the subset U with probability α and a vertex in the subset W with probability β. We determine Player 1's expected gains when she chooses a vertex in the subset U and when she chooses a vertex in the subset W. Again, we can force the two gains to be equal by solving for α and β knowing that $m\alpha + n\beta = 1$. This leads to the following two strategies.

Definition 6.10. Let the strategy λ_1 be a mixed strategy on $K_{m,n}$ where $\lambda_1 = (x_1, x_2, ..., x_n)$ and x_i, the probability of choosing the vertex v_i, is

$$x_i = \begin{cases} \alpha = \frac{mn-2n+1}{m^2n+mn^2+m+n-4mn}, & \text{if } v_i \in U \\ \beta = \frac{mn-2m+1}{m^2n+mn^2+m+n-4mn}, & \text{if } v_i \in W \end{cases} \tag{6.21}$$

for $1 \leq i \leq n$.

Definition 6.11. Let the strategy λ_2 be a mixed strategy on $K_{m,n}$ where $\lambda_1 = (y_1, y_2, ..., y_n)$ and y_j, the probability of choosing the vertex v_j, is

$$x_j = \begin{cases} \alpha = \frac{(n-1)^2}{m^2n+mn^2+m+n-4mn}, & \text{if } v_j \in U \\ \beta = \frac{m^2-2m+1}{m^2n+mn^2+m+n-4mn}, & \text{if } v_j \in W \end{cases} \tag{6.22}$$

for $1 \leq j \leq n$.

We consider the strategy λ_1 as a safe strategy for Player 1 on $K_{m,n}$ and the strategy λ_2 as an opposing strategy for Player 2 on $K_{m,n}$. This leads to the following results:

Theorem 6.12. *In the two-player safe game of Competitive Diffusion on the complete bipartite graph $K_{m,n}$, the safety value of Player 1 is*

$$\frac{(m-1)(n-1)(mn-1)}{m^2n + mn^2 + m + n - 4mn} \tag{6.23}$$

In order to prove this theorem, we will consider some lemmas on the guaranteed gain of Player 1 with the safe strategy λ_1 and the maximal gain of Player 1 when Player 2 has the opposing strategy λ_2.

Lemma 6.13. *The guaranteed gain of Player 1 with the strategy λ_1 on a complete bipartite graph $K_{m,n}$ is*

$$GGain(K_{m,n}, \lambda_1) = \frac{(m-1)(n-1)(mn-1)}{m^2n + mn^2 + m + n - 4mn}. \tag{6.24}$$

Proof. From Definition 2.9, we know that the guaranteed gain of Player 1 with the strategy λ_1 on $K_{m,n}$ is

$$GGain(K_{m,n}, \lambda_1) = \min_j Gain(K_{m,n}, \lambda_1, Z(v_j)) \tag{6.25}$$

where $1 \leq j \leq n$ and $Z(v_j)$ is the mixed strategy which chooses the vertex v_j with probability 1. Since all the edges between the vertices in the bipartition subsets U and W of $K_{m,n}$ are present, there is no distinction to be made between the vertices from a given subset. Thus, we only need to consider one vertex from the subset U and one vertex from the subset W as possible starting vertices for Player 2.

If Player 2 chooses a vertex $u_j \in U = \{u_1, u_2, ..., u_m\}$, the expected gain of Player 1 is

$$\begin{aligned} Gain(K_{m,n}, \lambda_1, Z(u_j)) = \alpha \sum_{i=1}^{m} Gain(K_{m,n}, Z(u_i), Z(u_j)) \\ + \beta \sum_{i=1}^{n} Gain(K_{m,n}, Z(w_i), Z(u_j)) \end{aligned} \tag{6.26}$$

where $\{w_1, w_2, ..., w_n\}$ are the vertices in the subset W and α and β are respectively the probability of choosing a vertex in the subset U and the probability of choosing a vertex in the subset W as defined in Definition 6.10. If the two players choose two different vertices in the same subset, the payoff to Player 1 is 1 since all the vertices of the other subset are neighbours to both chosen vertices and will turn grey. If the two players choose vertices in different subsets, then the payoff to Player 1 will be all the vertices in the subset with the chosen vertex of Player 2 except the one chosen by Player 2. Thus,

$$Gain(K_{m,n}, \lambda_1, Z(u_j)) = \alpha(0 + (m-1) \cdot 1) + \beta(n(m-1)). \tag{6.27}$$

Replacing the values of α and β from Definition 6.10 gives

$$Gain(K_{m,n}, \lambda_1, Z(u_j)) = \frac{(m-1)(n-1)(mn-1)}{m^2n + mn^2 + m + n - 4mn}. \tag{6.28}$$

In a similar manner, we can determine the expected gain of Player 1 when Player 2 chooses a vertex $w_j \in W = \{w_1, w_2, ..., w_n\}$ to be

$$Gain(K_{m,n}, \lambda_1, Z(w_j)) = \frac{(m-1)(n-1)(mn-1)}{m^2n + mn^2 + m + n - 4mn}. \tag{6.29}$$

Since both these gains are equal, this is the guaranteed gain of Player 1 with the strategy λ_1 on $K_{m,n}$. □

Lemma 6.14. *The maximal gain of Player 1 when Player 2 uses the opposing strategy λ_2 on a complete graph $K_{m,n}$ is*

$$MGain(K_{m,n}, \lambda_2) = \frac{(m-1)(n-1)(mn-1)}{m^2n + mn^2 + m + n - 4mn}. \tag{6.30}$$

Proof. From Definition 2.10, we know that the maximal gain of Player 1 against the strategy λ_2 for Player 2 is

$$MGain(K_{m,n}, \lambda_2) = \max_i Gain(K_{m,n}, Z(v_i), \lambda_2) \tag{6.31}$$

where $1 \le i \le n$. Again, because of the structure of $K_{m,n}$, we only need to consider one vertex from the subset U and one vertex from the subset W as possible starting vertices for Player 1.

If Player 1 chooses a vertex $u_i \in U = \{u_1, u_2, ..., u_m\}$, her expected gain is

$$\begin{aligned} Gain(K_{m,n}, u_i, \lambda_2) = \alpha &\sum_{j=1}^{m} Gain(K_{m,n}, Z(u_i), Z(u_j)) \\ + \beta &\sum_{j=1}^{n} Gain(K_{m,n}, Z(u_i), Z(w_j)) \end{aligned} \tag{6.32}$$

where $\{w_1, w_2, ..., w_n\}$ are the vertices in the subset W and α and β are respectively the probability of choosing a vertices in the subset U and the probability of choosing a vertex in the subset W as defined in Definition 6.11. Again, if the two players choose two different vertices in the same subset, the payoff to Player 1 is 1 while if the two players choose vertices in different subsets, the payoff to Player 1 is 1 less than the number of vertices in the subset chosen by Player 2. Thus,

$$Gain(K_{m,n}, Z(u_i), \lambda_2) = \alpha(0 + (m-1)) + \beta(n(n-1)). \tag{6.33}$$

Replacing the values of α and β from Definition 6.11 gives

$$Gain(K_{m,n}, Z(u_i), \lambda_2) = \frac{(m-1)(n-1)(mn-1)}{m^2n + mn^2 + m + n - 4mn}. \tag{6.34}$$

In a similar manner, we can determine the expected gain of Player 1 when she chooses a vertex $w_i \in W = \{w_1, w_2, ..., w_n\}$ to be

$$Gain(K_{m,n}, Z(w_i), \lambda_2) = \frac{(m-1)(n-1)(mn-1)}{m^2n + mn^2 + m + n - 4mn}. \tag{6.35}$$

Since these gains are equal, this is the maximal gain of Player 1 against the strategy λ_2 of Player 2. $\square$

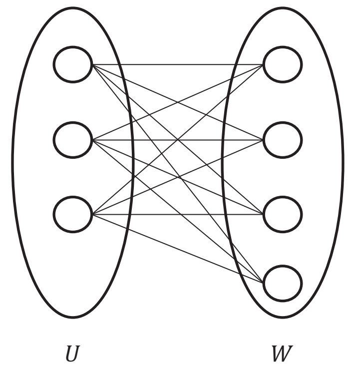


Figure 6.3: Example of a Complete Bipartite Graph, $K_{m,n}$ with $m = 3$ and $n = 4$.

We now prove Theorem 6.3.

Proof. **(Theorem 6.3)** This proof is essentially the same as the proof of Theorem 3.5 having the guaranteed gain from Lemma 6.13 and the maximal gain from Lemma 6.14 except that in this case, the bounds are equal. Thus, the safety value of Player 1 is exactly

$$\frac{(m-1)(n-1)(mn-1)}{m^2n + mn^2 + m + n - 4mn}. \tag{6.36}$$

□

Example 6.15. For the complete bipartite graph, $K_{m,n}$ represented in Figure 6.3 where $m = 3$ and $n = 4$, the safety value of Player 1 is 1.54.

Example 6.16. For the complete bipartite graph, $K_{m,n}$ with $m = 100$ and $n = 100$, the safety value of Player 1 is 50.00 while with $m = 50$ and $n = 150$ the safety value of Player 1 is 37.24.

A similar reasoning could be applied to get the safety value of Player 1 along with a safe strategy for complete multipartite graphs, graphs whose vertices can be partitioned into sets such that two vertices are adjacent if and only if they belong to different sets.

Chapter 7

Conclusion

The main goal of this thesis was to study the two-player safe game of Competitive Diffusion on trees. The guaranteed gain in the safe game is the minimal expected gain of a player if he adopts the corresponding safe strategy. Thus, safe strategies are a good approach in presence of uncertainty in opponents.

Consider, for example, the setting where Competitive Diffusion is a game between two rival companies who want to advertise their products throughout a social network. We can easily imagine companies that do not like taking risks, small companies that cannot afford the chance of not getting information about their product to at least a few buyers, or companies that think that their opponents are out to get them. In all these scenarios, a safe strategy can be a good option. For other companies, knowing the guaranteed gain with a certain safe strategy can be helpful in decision making. A company might realize that the expected gain assured with the safe strategy is satisfactory.

We started the study of the safe game of Competitive Diffusion in Chapters 3 and 4 with the two-player game on paths, spiders and complete trees. These are all special cases of trees. For each, we suggested safe strategies for Player 1 and compared the guaranteed gains to the safety values by evaluating their proximity to upper bounds obtained by considering opposing strategies for Player 2. In Chapter 5, we turned our focus to finding a general safe strategy for trees. Our approach leads to the Centroidal Safe Strategy Algorithm. The algorithm showed a good performance on a number of examples; the ones with thick branches at the centroid better than the ones with thin branches at the centroid. The algorithm perhaps could be improved by including positive probabilities on more vertices in the branches at the centroid or by slightly modifying the ordering of the branches and the distribution of the probabilities. Nevertheless, the algorithm seems a good base to begin the study of the safe game on bicentroidal trees in future work.

In Chapter 6, we discussed the safe game of Competitive Diffusion on cycles, complete graphs and complete bipartite graphs. These are simple and symmetrical examples of graphs which are not trees. Still, the safe strategies considered in Chapter 6 were optimal. Hence, our common approach of equalling expected gains of the players to suggest probabilities in strategies shows promise. The notion of the centroid, however, does not directly translate to graphs that do not have branches but there exists a generalization, also known as an accretion center (see [28]). Thus, strategies which have positive probabilities near this notion of centroid might be worth considering on graphs which are not trees. Moreover, there exists discussion on a decentralized approach to approximate the location of the centroid without the global view of the graph environment (see [8]). Using a local view of a graph could be useful for studying the centroid in complex graphs such as large social networks. Thus, this approach might be useful in generalizing the study of the safe game of Competitive Diffusion on social networks. Since the ultimate goal is to transfer the ideas to the Competitive Diffusion on realistic networks, the suggestions mentioned here are all of interest for future work on the safe game of Competitive Diffusion.

Lastly, changing the concept of grey vertices is probably the first thing that comes to mind when thinking of modifying the simple game model of Competitive Diffusion. It seems unreasonable that a user facing two choices of products, for example, remains indecisive rather than choosing one of the two products with a certain probability. However, in the conservative setting of the safe game, the grey vertices are not a bad concept. They restrict the payoff to Player 1 by eliminating all chances of her gaining vertices other than the ones she gets to influence before her opponents. This said, modifying the game model to include a probability of Player 1 gaining other vertices, would simply increase her safety value. The guaranteed gains that serve as lower bounds on the safety values for the Competitive Diffusion on the graphs presented, should to some extent be translatable to the new model. Moreover, considering models which include a probability of gaining the grey vertices would create large variations in the gains. In a tree, a vertex that turns grey usually blocks the access to a branch with a number of vertices. A player that gains this vertex would also gain the rest of the branch. Thus, the probability of gaining this large group of vertices would create fluctuations in the expected gains. Overall, the simple cascade model in the Competitive Diffusion remains a good conservative model for the safe game, a concept in game theory to which the newly decentralized game theoretic setting permits analysis.

Appendix A

Programming Codes

This appendix presents the programming codes used in Chapter 5 to assess the performance of the Centroidal Safe Strategy Algorithm 5.21.

A.1 Generating Centroidal Tree Examples: Method 1

This section presents the MATLAB® [19] programming code used to generate the examples of centroidal trees with the first method of randomization. Recall from Section 5.2.5.

Method 1:

- **Fixed parameters**: number of branches at the centroid, d, and number of vertices in the largest branch at the centroid, n_1.
- **Randomized parameters**: number of vertices in the other branches and lowest three weights in the branches.

The randomization in this method is obtained by a function in MATLAB® [19].

`randi([imin,imax],1)`:[20] returns an integer value drawn from the discrete uniform distribution on the interval [imin,imax].

The generated trees are written in a text file called "INPUT.txt" created in a specified directory.

```
%RANDOMIZATION OF CENTROIDAL TREES WITH METHOD 1.

%[INPUT]:

%Specify the text file and directory where the examples generated ...
    will be printed.
[Results,message]=fopen(fullfile('C:\Users\Celeste\Documents\...
    MATLAB','INPUT.txt'),'a');

%Specify the number of branches at the centroid wanted.
d=3;

%Specify the number of vertices in the largest branch at the ...
    centroid.
n1=100;

%Specify the number of examples of centroidal trees wanted.
NEx=20000;

%Loop that creates and prints the examples.
for q=1:NEx
    A=zeros(d,4);
    A(1,4)=n1;

    %[STEP 1]: Loop that randomizes the number of vertices in the ...
        branches 2 to d.
    for i=2:d
        %All the branches except the last one can have any number ...
            of vertices from 1 to n1.
        if i<d
            A(i,4)=randi([1 A(1,4)],1,1);
        %The number of vertices in the last branch must assure ...
            that the sum of the vertices on the branches 2 to d is ...
            at least n1.
        else if i==d
                s=A(1,4)-sum(A(2:d-1,4));
                if s>1
                    A(i,4)=randi([s A(1,4)],1,1);
```

```
                else
                    A(i,4)=randi([1 A(1,4)],1,1);
                end
            end
        end
    end

    %The total number of vertices is the sum of the number of ...
        vertices in the d branches +1 for the centroid.
    n=sum(A(:,4))+1;

    %[STEP 2]: Loop that randomizes the lowest three weights in ...
        the branches.
    for i=1:d
        %The first weight of the branch i, w1, is fixed to n−ni.
        A(i,1)=n−A(i,4);
        %If the branch i has at least 2 vertices, the second ...
            lowest weight, w2, is any integer drawn from the ...
            interval [w1+1,n−1]. Otherwise, the second lowest ...
            weight stays 0.
        if A(i,4)≥2
            A(i,4)=randi([n−A(i,4)+1 n−1],1,1);
        end
        %If the branch i has at least 3 vertices, the third lowest...
             weight, w3, is any integer drawn from the interval [w2...
            ,n−1] if w2−w1−1 is greater than or equal to n−w2, else...
             [w2+1,n−1]. Otherwise, the third lowest weight stays 0...
            .
        if A(i,4)≥3
            if A(i,2)−A(i,1)−1≥n−A(i,2)
                A(i,3)=randi([A(i,2) n−1],1,1);
            else A(i,3)=randi([A(i,2)+1 n−1],1,1);
            end
        end
    end
```

```
    %[OUTPUT]: The example is printed on a line in the text file '...
       INPUT.txt' with the format [n d w(u_1) w(t_1) w(s_1) ... w(...
       u_d) w(t_d) w(s_d)] where u_i, t_i and s_i are respectively...
        the vertices with the first, second and third lowest ...
       weights in the branch i.
    fprintf(Results,'%g ', n, d, A(:,1:3)');
end
fclose(Results);
```

A.2 Generating Centroidal Tree Examples: Method 2

This section presents the Maple™ programming code used to generate the examples of centroidal trees with the second method of randomization. Recall from Section 5.2.5.

Method 2:

- **Fixed parameters**: total number of vertices, n.
- **Randomized parameters**: tree structure.

The randomization in this method is obtained by a function of Maple™ in the GraphTheory[RandomGraphs] subpackage.

The generated trees are written in a text file called "INPUT.txt" created in a specified directory.

```
> #RANDOMIZATION OF CENTROIDAL TREES WITH METHOD 2.
>
> #Load Packages.
> with(GraphTheory):
> with(RandomGraphs):
>
> #[INPUT]:
>
> #Specify the directory and text file where the examples ...
   generated will be printed.
> currentdir("C:\\Users\\Celeste\\Documents\\MAPLE"):
> fd := fopen("INPUT.txt", APPEND):
>
```

```
> #Specify the number of vertices in the tree wanted.
> n := 100:
>
> #Specify the number of examples of centroidal trees wanted.
> NEx := 1000:
>
> #Loop that creates and prints the examples.
> for k from 1 to NEx do

      #[STEP 1]: Generate a random tree with n vertices and list...
          the vertices of the tree.
      T := RandomTree(n):
      V := {Vertices(T)[]}:

      #Create an array where the weights of the vertices in T ...
         will be stored.
      W := Array(1..1, 1..numelems(V)):

      #[STEP 2]: Loop that determines the weights of all the ...
         vertices in the tree.
      for i from 1 to numelems(V) do

          #Let G be the induced subgraph of T by all the ...
             vertices of T except the vertex i and let C be the ...
             list of connected components of G. The elements of ...
             C represent the branches of T at the vertex i.
          G := InducedSubgraph(T, V\{i}):
          C := {ConnectedComponents(G)[]}:

          #Count the number of edges in all the branches at the ...
             vertex i and determine the maximum. This is the ...
             weight of the vertex i which is stored in the ...
             vertex attribute "weight" and the array W.
          A := Array(1..1, 1..numelems(C)):
          for j from 1 to numelems(C) do
              A[1, j] := numelems(C[j]):
          end do:
          SetVertexAttribute(T, i, "weight" = max(A)):
          W[1, i] := GetVertexAttribute(T, i, "weight"):
```

```
    end do:

    #[STEP 3:] Determine the centroid of the tree T, i.e. the ...
       vertices with the minimal weight. The variable position...
        will hold the vertex number of the centroid and the ...
       flag is used to display if a tree is bicentroidal.
    position := 0:
    Flag := "Ok":

    for i from 1 to numelems(V) do
        if W[1, i] = min(W) and position = 0 then
            position := i:
        elif W[1, i] = min(W) and position <> 0 then
            Flag := "Bicentroidal Tree" :
        end if:
    end do:

    #[STEP 4]: Determine the number of branches at the ...
       centroid, d, and the lowest three weights in the ...
       branches at the centroid, stored in an array M. If a ...
       branch has only 1 or 2 vertices, only one or two lowest...
        weights are stored for that branch.
    G := InducedSubgraph(T, V\{position}):
    C := {ConnectedComponents(G)[]}:
    d := numelems(C):
    M := Array(1..d, 1..3):
    for i from 1 to d do
        B := Array(1..numelems(C[i])):
        for j from 1 to numelems(C[i]) do
            B[j] := GetVertexAttribute(T, C[i][j], "weight"):
        end do:
        test := min(numelems(B), 3):
        M[i, 1..test] := sort(B)[1..test]:
    end do:
```

```
        #[OUTPUT]: If the example is a centroidal tree, it is ...
            printed on a line in the text file 'INPUT.txt' with the...
             format [n d w(u_1) w(t_1) w(s_1) ... w(u_d) w(t_d) w(s...
            _d)] where u_i, t_i and s_i are respectively the ...
            vertices with the first, second and third lowest ...
            weights in the branch i.  If the example is a ...
            bicentroidal tree, the example is disregarded and the ...
            counter is reduced by 1.
        if Flag = "Ok" then
            fprintf(fd, "%g ", n, d):
            fprintf(fd, "%{n}g \n", M):
        elif Flag="Bicentroidal Tree" then
            k:=k-1:
        end if:
end do:
fclose(fd):
```

A.3 Centroidal Safe Strategy Algorithm

This section presents the MATLAB® [19] programming code corresponding to the CSS Algorithm 5.21. The code reads the "INPUT.txt" files from Sections A.1 and A.2, i.e. the centroidal tree examples created by method 1 and 2. The results are then written in a text file called "RESULTS.txt" in a specified directory.

```
%CENTROIDAL SAFE STRATEGY ALGORITHM.

%[INPUT]: Specify the directory and the text file where the ...
    examples are read from.
[INPUT,message]=fopen(fullfile('C:\Users\Celeste\Documents\MATLAB'...
    ,'INPUT.txt'),'r');

%Read a line from text file.
tline = fgetl(INPUT);

%While the line read is not empty.
while tline ~= -1

```

```
%[STEP 0]: Convert the information from the text file to n, ...
    the number of vertices, d, the number of branches at the ...
    centroid, and A, a matrix with the lowest three weights in ...
    each branch.
temp=textscan(tline,'%f ');
n=temp{1}(1,1);
d=temp{1}(2,1);
A=zeros(d,6);
for i=1:d
    A(i,1:3)=temp{1}(3+3*(i-1):5+3*(i-1),1)';
    A(i,6)=n-A(i,1);
end

%[STEP 1a]: Loop that determines the type of each branch and ...
    calculates its criterion. If a branch has less than 3 ...
    vertices, then by convention, its criterion is zero and the...
     branch is labelled as a thick branch. A thin branch is ...
    identified by the number 1, a medium branch by the number 2...
     and a thick branch by the number 3.
for i=1:d
    if A(i,6)<3
        A(i,4)=3;
        A(i,5)=0;
        else if A(i,2)>=n-A(i,1)+(A(i,1))^2/n
            A(i,4)=3;
            A(i,5)=n-A(i,1);
            else if A(i,2)<n-A(i,1)+(A(i,1))^2/n && A(i,3)>=n-...
                A(i,2)+(A(i,2)^2+(A(i,2)-A(i,1))^2)/(n+A(i,2)-A...
                (i,1))
                A(i,4)=2;
                A(i,5)=((n-A(i,2))/n)*(n-A(i,1))+(A(i,2)/n)*(n...
                    -A(i,2));
                else if A(i,2)<n-A(i,1)+(A(i,1))^2/n && A(i,3)...
                    <n-A(i,2)+(A(i,2)^2+(A(i,2)-A(i,1))^2)/(n+A...
                    (i,2)-A(i,1))
                    A(i,4)=1;
```

```
                        A(i,5)=(A(i,2)*(n-A(i,2))*(n^2-n*A(i,3)-A(...
                            i,3)*A(i,2)+A(i,2)^2+2*A(i,3)*A(i,1)-A(...
                            i,2)*A(i,1)))/(n*A(i,2)*(n-A(i,3))+A(i...
                            ,1)*A(i,2)*(-n+A(i,3)+A(i,2))+(n-A(i,2)...
                            )*A(i,1)^2);
                    end
                end
            end
        end
    end

    %[STEP 1b]: Order the branches at the centroid based on their ...
        criterion.
    B=sortrows(A,-5);

    j=1;
    SGain=0;
    EGain=0;
    SProbAlpha=0;
    Alpha=0;

    %[STEP 2]: While there are branches remaining and the ...
        criterion of the next branch is greater than the current ...
        expected gain, add positive probabilities on the next ...
        branch and calculate the new expected gain dependently on ...
        the new branch being thick, medium or thin.
    while j<=d && B(j,5)>EGain
        %[STEP 3a]: Solve $\alpha+\sum_{i=1}^{j}(\beta_j+\gamma_j+\delta_j)=1$ for $\alpha$.
        if B(j,4)==1
            SProbAlpha=SProbAlpha+((n-B(j,2))*(B(j,1)*(n-B(j,3))+(...
                B(j,2)-B(j,3))*(B(j,2)-B(j,1))))/((n-B(j,3))*(n-B(j...
                ,1))*(n-B(j,2))+B(j,3)*B(j,2)*(B(j,3)-B(j,2)))+B(j...
                ,2)/(n-B(j,2))*((n-B(j,2))*(B(j,1)*(n-B(j,3))+(B(j...
                ,2)-B(j,3))*(B(j,2)-B(j,1))))/((n-B(j,3))*(n-B(j,1)...
                )*(n-B(j,2))+B(j,3)*B(j,2)*(B(j,3)-B(j,2)))+B(j,3)/...
                (n-B(j,3))*B(j,2)/(n-B(j,2))*((n-B(j,2))*(B(j,1)*(n...
                -B(j,3))+(B(j,2)-B(j,3))*(B(j,2)-B(j,1))))/((n-B(j...
                ,3))*(n-B(j,1))*(n-B(j,2))+B(j,3)*B(j,2)*(B(j,3)-B(...
                j,2)))+(B(j,2)-B(j,1))/(n-B(j,3));
```

```
            else if B(j,4)==2
                    SProbAlpha=SProbAlpha+B(j,1)/(n-B(j,1))+B(j,2)...
                        /(n-B(j,2))*B(j,1)/(n-B(j,1));
                else if B(j,4)==3
                    SProbAlpha=SProbAlpha+B(j,1)/(n-B(j,1));
                end
            end
        end
        Alpha=1/(1+SProbAlpha);
        %[STEP 3b]: With the strategy σ_j , determine the expected ...
            gain of Player 1 if Player 2 chooses the centroid.
        if B(j,4)==1
            SGain=SGain+((n-B(j,2))*(B(j,1)*(n-B(j,3))+(B(j,2)-B(j...
                ,3))*(B(j,2)-B(j,1))))/((n-B(j,3))*(n-B(j,1))*(n-B(...
                j,2))+B(j,3)*B(j,2)*(B(j,3)-B(j,2)))*(n-B(j,1))+B(j...
                ,2)/(n-B(j,2))*((n-B(j,2))*(B(j,1)*(n-B(j,3))+(B(j...
                ,2)-B(j,3))*(B(j,2)-B(j,1))))/((n-B(j,3))*(n-B(j,1)...
                )*(n-B(j,2))+B(j,3)*B(j,2)*(B(j,3)-B(j,2)))*(n-B(j...
                ,2))+B(j,3)/(n-B(j,3))*B(j,2)/(n-B(j,2))*((n-B(j,2)...
                )*(B(j,1)*(n-B(j,3))+(B(j,2)-B(j,3))*(B(j,2)-B(j,1)...
                )))/((n-B(j,3))*(n-B(j,1))*(n-B(j,2))+B(j,3)*B(j,2)...
                *(B(j,3)-B(j,2)))*(n-B(j,2))+(B(j,2)-B(j,1))/(n-B(j...
                ,3))*(n-B(j,2));
            else if B(j,4)==2
                SGain=SGain+B(j,1)/(n-B(j,1))*(n-B(j,1))+B(j,2)/(n...
                    -B(j,2))*B(j,1)/(n-B(j,1))*(n-B(j,2));
                else if B(j,4)==3
                    SGain=SGain+B(j,1)/(n-B(j,1))*(n-B(j,1));
                end
            end
        end
        EGain=Alpha*SGain;
        j=j+1;
    end

    %[OUTPUT]: Print the guaranteed gain of Player 1 in the text ...
        file 'RESULTS.txt' in the specified directory.
    [Results,message]=fopen(fullfile('C:\Users\Celeste\Documents\...
        MATLAB','RESULTS.txt'),'a');
```

```
    fprintf(Results,'%g ',EGain);
    fclose(Results);

    %Read the next example from the text file 'INPUT.txt'.
    tline = fgetl(INPUT);
end
fclose(INPUT);
```

A.4 Centroidal Opposing Strategy Algorithm

This section presents the MATLAB® [19] programming code corresponding to the COS Algorithm 5.32. The code reads the "INPUT.txt" files from Sections A.1 and A.2, i.e. the centroidal tree examples created by method 1 and 2. The results are then written in a text file called "RESULTS.txt" in a specified directory.

```
%CENTROIDAL OPPOSING STRATEGY ALGORITHM:

%[INPUT]: Specify the directory and the text file where the ...
    examples are read from.
[INPUT,message]=fopen(fullfile('C:\Users\Celeste\Documents\MATLAB'...
    ,'INPUT.txt'),'r');

%Read a line from text file.
tline = fgetl(INPUT);

%While the line read is not empty.
while tline ~= -1

    %[STEP 0a]: Convert the information from the text file to n, ...
        the number of vertices, d, the number of branches at the ...
        centroid, and A, a matrix with the lowest three weights in ...
        each branch.
    temp=textscan(tline,'%f ');
    n=temp{1}(1,1);
    d=temp{1}(2,1);
    A=zeros(d,6);
    for i=1:d
```

```
        A(i,1:3)=temp{1}(3+3*(i−1):5+3*(i−1),1)';
        A(i,6)=n−A(i,1);
    end

    %[STEP 0b]: Loop that determines the type of each branch and ...
       calculates its criterion. If a branch has less than 3 ...
       vertices, then by convention, its criterion is zero and the...
        branch is labelled as a thick branch. A thin branch is ...
       identified by the number 1, a medium branch by the number 2...
        and a thick branch by the number 3.
    for i=1:d
        if A(i,6)<3
            A(i,4)=3;
            A(i,5)=0;
            else if A(i,2)>=n−A(i,1)+(A(i,1))^2/n
                A(i,4)=3;
                A(i,5)=n−A(i,1);
                else if A(i,2)<n−A(i,1)+(A(i,1))^2/n && A(i,3)>=n−...
                    A(i,2)+(A(i,2)^2+(A(i,2)−A(i,1))^2)/(n+A(i,2)−A...
                    (i,1))
                    A(i,4)=2;
                    A(i,5)=((n−A(i,2))/n)*(n−A(i,1))+(A(i,2)/n)*(n...
                        −A(i,2));
                    else if A(i,2)<n−A(i,1)+(A(i,1))^2/n && A(i,3)...
                        <n−A(i,2)+(A(i,2)^2+(A(i,2)−A(i,1))^2)/(n+A...
                        (i,2)−A(i,1))
                        A(i,4)=1;
                        A(i,5)=(A(i,2)*(n−A(i,2))*(n^2−n*A(i,3)−A(...
                            i,3)*A(i,2)+A(i,2)^2+2*A(i,3)*A(i,1)−A(...
                            i,2)*A(i,1)))/(n*A(i,2)*(n−A(i,3))+A(i...
                            ,1)*A(i,2)*(−n+A(i,3)+A(i,2))+(n−A(i,2)...
                            )*A(i,1)^2);
                    end
                end
            end
        end
    end

```

```
    %[STEP 1]:Include some positive probabilities in the opposing ...
        strategy of Player 2 on some vertices adjacent to the ...
        centroid.

    %[STEP 1a]: Order the branches at the centroid based on their ...
        number of vertices.
    C=sortrows(A,-6);

    l=1;
    MEGain=0;

    %[STEP 1b-1c]: While there are branches remaining and the ...
        number of vertices in the next branch is greater than the ...
        current expected gain, add probability on the first vertex ...
        of the next branch and calculate the new expected gain.
    while l<=d && C(l,6)>MEGain

        %[STEP 1b.0]: Let the opposing strategy of Player 2 be a ...
            probability of 'a' on the centroid and 'b1','b2',...,'...
            bl' on the first vertices of the l branches.
        syms a;
        b=sym('b',[d,1]);
        var=[a; b(1:l,1)];

        %[STEP 1b.1]: Calculate the payoffs to Player 1 when she ...
            and Player 2 choose the vertices amongst the centroid, ...
            c, and the first vertices of the l branches, u1, u2...
            ,..., ul.
        M=zeros(l+1,l+1);
        M(1,:)=[0 C(1:l,1)'];
        for i=2:l+1
            M(i,:)=[repmat(n-C(i-1,1),1,i-1), 0 , repmat(n-C(i...
                -1,1),1,l+1-i)];
        end

        %[STEP 1b.2]: List the expected gains of Player 1 against ...
            the opposing strategy of Player 2 when she chooses the ...
            vertices, c, u1, u2,...,ul.
        equations=sym('eqn',[1,l+1]);
```

```
        equations(1,1)=a+sum(b(1:l,1))==1;
        for i=1:l
            equations(1,i+1)=M(i,:)*var==M(i+1,:)*var;
        end

        %[STEP 1c.1]: Equal the expected gains of Player 1 by ...
            solving for 'a', 'b1', 'b2',...,'bl', knowing a+b1+b2...
            +...+bl=1.
        ListEqns=num2cell(equations);
        ListVar=num2cell(var);
        V=solve(ListEqns{:},ListVar{:});
        Prob=zeros(l+1,1);
        Prob(1,1)=V.a;
        for i=1:l
            Prob(i+1,1)=V.(sprintf('b%d',i));
        end

        %[STEP 1c.2]: Calculate the expected gain of Player 1 when...
             she chooses the centroid.
        MEGain=M(1,:)*Prob;

        l=l+1;
    end

    %[STEP 2]: Include some positive probabilities in the opposing...
         strategy of Player 2 on some vertices at distance 2 from ...
        the centroid.

    %[STEP 2a.1]: Calculate the expected gains of Player 1, when ...
        she chooses a second vertex with the lowest weight in the ...
        branches 1 to l-1.
    OGain=zeros(1,l-1);
    for i=1:l-1
        OGain(1,i)=(Prob(1,1)+Prob(i+1,1))*(n -C(i,2))+sum(Prob...
            ([2:i,i+2:l],1))*(n-C(i,1));
    end


    IMEGain=MEGain;
```

```
    IOGain=OGain;
    IProb=Prob;

    %[STEP 2a.2]: Order the l−1 branches on which Player 2 has a ...
        positive probability on the vertex with the lowest weight ...
        by the expected gain of Player 1 when she chooses the ...
        second vertex with the lowest weight in the branches.
    D=[C, [OGain, zeros(1,d−l+1)]'];
    D=sortrows(D,−7);
    h=1;

    %[STEP 2b−2c]: While there are branches on which Player 2 has ...
        a positive probability on the vertex with the lowest weight...
         remaining, the expected gain of Player 1 when she chooses ...
        the second vertex with the lowest weight in the next branch...
         is greater than the current expected gain and the strategy...
         for Player 2 is still legal, add probabilities on the ...
        second vertex of the next branch and calculate the new ...
        expected gain.
    while h<=l−1 && D(h,7)>MEGain+0.0001 && sum(Prob<0)==0

        %[STEP 2b.0.1]: Save the maximal gain of P1 against the ...
            strategy of Player 2 in the case that adding ...
            probability on the second vertex with the lowest weight...
             in the next branch creates an invalid strategy for ...
            Player 2, i.e. a strategy with negative probabilities.
        IMEGain=MEGain;
        IOGain=OGain;
        IProb=Prob;

        %[STEP 2b.0.2]: Let the opposing strategy of Player 2 be a...
             probability of 'a' on the centroid, 'b1', 'b2',...,'b{...
            l−1}' on the first vertices of l−1 branches and 'g1','...
            g2',...,'gh' on the second vertex of h branches.
        syms a;
        b=sym('b',[d,1]);
        g=sym('g',[d,1]);
        var=[a; b(1:l−1,1);g(1:h)];

```

```
%[STEP 2b.1]: Calculate the payoffs to Player 1 when she ...
    and Player 2 choose vertices amongst the centroid, the ...
    first vertices of the l-1 branches and the second ...
    vertices of the h branches.
M=zeros(l-1+h+1,l-1+h+1);
M(1,:)=[0 D(1:l-1,1)' D(1:h,1)'];
for i=2:l-1+1
   if h-(i-1)>=0
        test=1;
        test2=i-2;
        else if h-(i-1)<0
            test=0;
            test2=h;
        end
    end
M(i,:)=[repmat(n-D(i-1,1),1,i-1), 0 , repmat(n-D(i-1,1),1,...
    l-1+1-i),D(1:test2,1)', repmat(D(i-1,2),1,test) , D(i:h...
    ,1)' ];
end
for i=l-1+2:l+h
    M(i,:)=[repmat(n-D(i-(l-1+1),2),1,1),repmat(n-D(i-(l...
        -1+1),1),1,i-(l-1+2)),repmat(n-D(i-(l-1+1),2),1,1),...
        repmat(n-D(i-(l-1+1),1),1,2*l-i-1),repmat(n-D(i-(l...
        -1+1),1),1,i-(l-1+2)),0,repmat(n-D(i-(l-1+1),1),1,l...
        +h-i)];
end

%[STEP 2b.2]: List the expected gains of Player 1 against ...
    the opposing strategy of Player 2 when she chooses the ...
    centroid, the first vertices of the l-1 branches and ...
    the second vertices of the h branches.
equations=sym('eqn',[1,l+h]);
equations(1,1)=a+sum(b(1:l-1,1))+sum(g(1:h,1))==1;
for i=1:l+h-1
    equations(1,i+1)=M(i,:)*var==M(i+1,:)*var;
end

```

```
        %[STEP 2c.1]: Equal the expected gains of Player 1 by ...
            solving for 'a','b1','b2',...,'b{l−1}','g1','g2',...,'...
            gh', knowing a+b1+b2+...+b{l−1}+g1+g2+...+gh=1.
        ListEqns=num2cell(equations);
        ListVar=num2cell(var);
        V=solve(ListEqns{:},ListVar{:});
        Prob=zeros(l+h,1);
        Prob(1,1)=V.a;
        for i=1:l−1
            Prob(i+1,1)=V.(sprintf('b%d',i));
        end
        for i=1:h
            Prob(l−1+i+1,1)=V.(sprintf('g%d',i));
        end
        MEGain=M(2,:)*Prob;

        %[STEP 2c.2]: Calculate the expected gains of Player 1 ...
            when she chooses a second vertex with the lowest weight...
             in the branches 1 to l−1.
        OGain=zeros(1,l−1);
        for i=1:l−1
            if h−i>=0
                test=l+i;
                else if h−i<0
                    test=l+h+1;
                end
            end
            OGain(1,i)=(Prob(1,1)+Prob(i+1,1))*(n−D(i,2))+sum(Prob...
                ([2:i,i+2:l,l+1:test−1,test+1:l+h],1))*(n−D(i,1));
        end
        D(:,7)= [OGain, zeros(1,d−l+1)]';

        h=h+1;
    end

```

```
%[STEP 3]: Calculate the maximal gain of Player 1 against the ...
    resulting opposing strategy of Player 2. If h = 0 or h > 0 ...
    and all the probabilities in the resulting opposing ...
    strategy for Player 2, are positive, calculate the maximal ...
    gain of Player 1 against the opposing strategy of Player 2 ...
    which has positive probabilities on the second vertices of ...
    0 or h-1 branches. If h > 0 and some negative probabilities...
     exists, calculate the maximal gain of Player 1 against the...
     opposing strategy of Player 2 which has positive ...
    probabilities on the second vertices in h-2 branches.

if sum(Prob<0)==0
    Flag=0;
else
    Flag=1;
    Prob=[Iprob;0];
end

%[STEP 3a]: Calculate the expected gains of Player 1 on ...
    another vertex in a branch where Player 2 has positive ...
    probabilities on the first and second vertices with lowest ...
    weights.
if Flag==0
    O2Gain=zeros(1,h-1);
    for i=1:h-1
        O2Gain(1,i)=Prob(1,1)*(n-D(i,2))+(Prob(i+1,1)+Prob(l+i...
            ,1))*(n-D(i,3))+sum(Prob([2:i,i+2:l,l+1:l+i-1,l+i...
            +1:l+h-1],1))*(n-D(i,1));
    end
else
    O2Gain=zeros(1,h-2);
    for i=1:h-2
        O2Gain(1,i)=Prob(1,1)*(n-D(i,2))+(Prob(i+1,1)+Prob(l+i...
            ,1))*(n-D(i,3))+sum(Prob([2:i,i+2:l,l+1:l+i-1,l+i...
            +1:l+h-1],1))*(n-D(i,1));
    end
end
```

```
    %[STEP 3b]: Calculate the expected gains of Player 1 on the ...
        vertices with the lowest weight in branches on which Player...
         2 does not assign any positive probabilities.
    O3Gain=zeros(1,d-l+1);
    for i=1:d-l+1
        O3Gain(1,i)=(sum(Prob(1:l,1)))*(n-  D(l-1+i,1))+Prob(l+1:l...
            +h-1,1)'*D(1:h-1,1);
    end

    %[OUTPUT]: Print the maximal gain of Player 1 against the ...
        resulting opposing strategy of Player 2 in the text file '...
        RESULTS.txt' in the specified directory.
    [Results,message]=fopen(fullfile('C:\Users\Celeste\Documents\...
        MATLAB','RESULTS.txt'),'a');
    if Flag==0
        fprintf(Results,'%g ',max([MEGain,OGain,O2Gain,O3Gain]));
        else
            fprintf(Results,'%g ',max([IMEGain,IOGain,O2Gain,...
                O3Gain]));
    end
    fclose(Results);

    %Read the next example from the text file 'INPUT.txt'.
    tline = fgetl(INPUT);
end
fclose(INPUT);
```

Bibliography

[1] Noga Alon, Michal Feldman, Ariel D. Procaccia, and Moshe Tennenholtz. A note on competitive diffusion through social networks. *Information Processing Letters*, 110(6):221 – 225, 2010.

[2] Sayan Bandyapadhyay, Aritra Banik, Sandip Das, and Hirak Sarkar. Voronoi game on graphs. In *WALCOM: Algorithms and Computation*, volume 7748 of *Lecture Notes in Computer Science*, pages 77–88. Springer Berlin Heidelberg, 2013.

[3] E. N. Barron. *Game Theory, An Introduction*. Wiley-Interscience, Hoboken, N.J, 2008.

[4] Allan Borodin, Yuval Filmus, and Joel Oren. Threshold models for competitive influence in social networks. In *Internet and Network Economics*, volume 6484 of *Lecture Notes in Computer Science*, pages 539–550. Springer Berlin Heidelberg, 2010. "and references therein.".

[5] Daniel Boudreau, Jeannette Janssen, Richard Nowakowski, and Elham Roshanbin. Safe strategies for competitive diffusion in social networks. Manuscript in Preparation.

[6] Joshua R. Davis, Zachary Goldman, Elizabeth N. Koch, Jacob Hilty, David Liben-Nowell, Alexa Sharp, Tom Wexler, and Emma Zhou. Equilibria and efficiency loss in games on networks. *Internet Mathematics*, 7(3):178–205, 2011.

[7] Christoph Dürr and Nguyen Kim Thang. Nash equilibria in Voronoi games on graphs. In *Algorithms ESA 2007*, volume 4698 of *Lecture Notes in Computer Science*, pages 17–28. Springer Berlin Heidelberg, 2007.

[8] Antoine Dutot, Damien Olivier, and Guilhelm Savin. Centroids : a decentralized approach. In *ECCS - European Conference on Complex System*, Vienna, Autriche, September 2011.

[9] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets : Reasoning about a Highly Connected World*. Cambridge University Press, New York, 2010.

[10] Sanjeev Goyal and Michael Kearns. Competitive contagion in networks. In *Proceedings of the 44th Symposium on Theory of Computing*, STOC '12, pages 759–774, New York, NY, USA, 2012. ACM.

[11] Nicole Immorlica, Jon Kleinberg, Mohammad Mahdian, and Tom Wexler. The role of compatibility in the diffusion of technologies through social networks. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, EC '07, pages 75–83, New York, NY, USA, 2007. ACM.

[12] Matthew O. Jackson. *Social and Economic Networks*. Princeton University Press, Princeton, NJ, 2008.

[13] Andy N.C. Kang and David A. Ault. Some properties of a centroid of a free tree. *Information Processing Letters*, 4(1):18 – 20, 1975.

[14] David Kempe, Jon Kleinberg, and Éva 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 '03, pages 137–146, New York, NY, USA, 2003. ACM. "and references therein.".

[15] Donald E. Knuth. *The Art of Computer Programming*. Addison-Wesley, Reading, Mass, 1997.

[16] *Maple 16*. Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario, 2012.

[17] Maplesoft a division of Waterloo Maple Inc. Maple User Manual. GraphTheory[RandomGraphs][RandomTree], (2005-2013). `http://www.maplesoft.com/support/help/Maple/view.aspx?path=GraphTheory/RandomGraphs/RandomTree`, Date accessed: January 23rd, 2014.

[18] *Mathematica 9.0*. Wolfram Research Inc., Champaign, Illinois, 2012.

[19] *MATLAB and Statistics Toolbox Version 8.0.0.783 (R2012b)*. The MathWorks, Inc., Natick, Massachusetts, 2012.

[20] The MathWorks Inc. R2013b Documentation.[randi], (2013). `http://www.mathworks.com/help/matlab/ref/randi.html`, Date accessed: January 23rd, 2014.

[21] Marios Mavronicolas, Burkhard Monien, Vicky G. Papadopoulou, and Florian Schoppmann. Voronoi games on cycle graphs. In *Mathematical Foundations of Computer Science 2008*, volume 5162 of *Lecture Notes in Computer Science*, pages 503–514. Springer Berlin Heidelberg, 2008.

[22] Sandra L. Mitchell. Another characterization of the centroid of a tree. *Discrete Mathematics*, 24(3):277 – 280, 1978.

[23] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

[24] John F. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[25] Chen Ning. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics*, 23(3):1400 – 1415, 2009.

[26] Jen-Ling Shang and Chiang Lin. Spiders are status unique in trees. *Discrete Mathematics*, 311(1011):785 – 791, 2011.

[27] Sunil Simon and Krzysztof R. Apt. Choosing products in social networks. In *Internet and Network Economics*, volume 7695 of *Lecture Notes in Computer Science*, pages 100–113. Springer Berlin Heidelberg, 2012.

[28] Peter J. Slater. Accretion centers: A generalization of branch weight centroids. *Discrete Applied Mathematics*, 3(3):187 – 192, 1981.

[29] Lucy Small and Oliver Mason. Information diffusion on the iterated local transitivity model of online social networks. *Discrete Applied Mathematics*, 161(1011):1338 – 1344, 2013.

[30] Lucy Small and Oliver Mason. Nash equilibria for competitive information diffusion on trees. *Information Processing Letters*, 113(7):217 – 219, 2013.

[31] Reiko Takehara, Masahiro Hachimori, and Maiko Shigeno. A comment on pure-strategy Nash equilibria in competitive diffusion games. *Information Processing Letters*, 112(3):59 – 60, 2012.

[32] Sachio Teramoto, Erik D. Demaine, and Ryuhei Uehara. The Voronoi game on graphs and its complexity. *Journal of Graph Algorithms and Applications*, 15(4):485–501, 2011.

[33] Paul R. Thie and Gerard E. Keough. *An Introduction to Linear Programming and Game Theory*. Wiley-Interscience, Hoboken, N.J, 2008.

[34] Vasileios Tzoumas, Christos Amanatidis, and Evangelos Markakis. A game-theoretic analysis of a competitive diffusion process over social networks. In *Internet and Network Economics*, volume 7695 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2012.