MERGER DYNAMICS IN THREE AGENT GAMES

by

Rüştü Derici

B.S., Information Technologies, Işık University, 2005

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computational Science & Engineering
Boğaziçi University

2009

ACKNOWLEDGEMENTS

I would like to thank with my utmost gratitude to my thesis supervisor Assoc. Prof. Tonguç Rador for his great guidance, invaluable teachings, patience and endless support.

Many thanks and appreciation to my supervisors Prof. M. Levent Kurnaz, Prof. Viktorya Aviyente, Assist. Prof. Ali Ecder for their sincere attention and greatest effort on solving all difficulties I confronted during my graduate study.

Together with Assoc. Prof. Tonguç Rador and Prof. M. Levent Kurnaz, I would like to thank to Asist. Prof. Nazım Ziya Perdahçı for their participation in my thesis committee.

I would like to thank to my family, especially my sister Özlem Derici for the careful reading of the thesis, and my father Muharrem Derici for his endless support and patience by all means.

ABSTRACT

MERGER DYNAMICS IN THREE AGENT GAMES

The thesis analyses the dynamics of merger in three agent games. The purpose of the study is to find out how merger affects the social structure.

To begin with the definition of a game, it is a series of competitions. In these competitions three players are randomly picked and they compete against each other with a given winning probability with respect to their actual points. In competitive games this means that the player with the highest score is favored.

In our model, there are two types of competitions with different probability sets. First is a competitive game and we define the second as merger game. Players act separately in competitive games. In merger games two players combine their points and act as a single player against the third. This would make sense if the players that merge increase their winning chance.

The merger is realized by resolving three agent games in terms of two agent mini tournaments. In a microscopic tournament, all players that participate in the microscopic competition will play a two agent game against each other and winning will be determined by the maximum wins in the mini tournament. Winner will gain one point.

Defining merger in three agent games doesn't violate the competitive structure of the game. Meanwhile it yields sub-societies among the total hierarchy.

ÖZET

ÜÇ KATILIMCILI OYUNLARDA BİRLEŞME DİNAMİKLERİ

Bu çalışmada üç katılımcılı oyunlarda birleşme dinamikleri incelenmiştir. Çalışmanın amacı, birleşmenin sosyal yapı üzerindeki etkilerini bulmaktır.

Oyunun tanımı, müsabakalar dizisi olarak belirtilmektedir. Bu müsabakalarda üç oyuncu rastgele seçilir ve seçilen oyuncular varolan puanlarına dayalı olarak kendilerine atanan kazanma olasılıkları ile birbirlerine karşı yarışırlar. Rekabetçil oyunlarda bu durum yüksek puanlı oyuncunun avantajlı olduğu anlamına gelir.

Bizim modelimizde, birbirinden farklı olasılık kümelerine sahip iki tip müsabaka vardır. Birincisi rekabetçil müsabakalardır. İkincisini ise birleşme oyunları olarak tanımladık. Bu iki oyun arasındaki fark, rekabetçil oyunlarda oyuncular tek başlarına oynarken, birleşme oyunlarında iki oyuncu puanlarını birleştirir ve üçüncü oyuncuya karşı tek bir oyuncu gibi darvanırlar. Bu hareket, birleşmeyi gerçekleştiren oyuncular kazanma olasılıklarını arttırdığında anlamlı olacaktır.

Birleşme, üç katılımcılı oyunları iki katılımcılı mini turnuvalara çözümlenerek gerçeklenir. Bir mini turnuvada, bütün katılımcılar birbirleri ile iki katılımcılı bir oyun oynarlar, ve bu mini oyunların çoğunu kazanan, oynanan müsabakayı da kazanarak puanını bir arttırır.

Üç katılımcılı oyunlarda birleşmenin tanımlanması, oyunun rekabetçil yapısına karşı gelmemekle birlikte, bütün düzen içerisinde alt topluluklar oluşturmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

f_x	fraction of players that have x points over total players
F_x	cumulative distribution of the points
$\overline{F}$	The value of F at $z = 1/2$
x, y, z	ordered points of the competitors from highest to lowest respectively
N	Total number of players in the whole society participating a game
n	number of players that participate to a single competition
P_{max}	Maximum points that a player may theoretically obtain in a particular time.
p, t, q	probability values for a no-merger game.
$\overline{p}, \overline{t}, \overline{q}$	probability values of the players playing a merging competition
s	probability modifier variable in generalization of the game for n+1 agents
v	the shock speed
$W_{x,y,z}$	Assigned winning probabilities for selected players in a competition
z	points normalized by time
z_l	leftmost shock location in F(z) - z graph
z_r	rightmost shock location in F(z) - z graph
z^*	rightmost location of the left bunch of a shock in F(z) - z graph
α	probability modifier constant in generalization of the game for n+1 agents, which is dependant to s
σ	length of the score region over the total score shared by them
ω	ratio of the players with lowest score versus highest scores
τ	time variable

1. INTRODUCTION

Competitive games and self-organizing hierarchies have been studied in many different areas. In 1998, Marsili and Zhang presented an approach to explain Zipf's law of city distribution [1], which was introduced at 1949. Opinion dynamics [2], sportive competitions, [3] and emergence of hierarchies and dynamics of multi-player games [4, 5, 6, 7, 8] are also examples of such studies.

These studies typically observe macroscopic phenomena with microscopic agent to agent interactions. Our study is based on the three agent game model which was studied in detail by Mungan and Rador in 2007[8]. In that model, game is defined as a series of competitions in a large group of players. Three players that participate in a competition are randomly selected and they compete against each other with a probability distribution based on the ordering of their points [8]. The resulting regimes depend on the defined probability set.

We studied a case in which there are two probability sets effecting the regime. By defining two types of competitions with different probability sets, we introduced a merger model where players with lower points gather their points and act like a single stronger player against the favored one under the condition that all merging players benefit from this action. For a better understanding of the problem we observed the models with one probability set.

In *Chapter 2*, two-agent model is observed and explained.

In *Chapter 3*, three agent model is studied and the regimes that came out of the model are observed.

In *Chapter 4*, a general solution to n+1 agent game is analyzed and an analytically solvable approach for this model is introduced.

In *Chapter 5*, the dynamics of merge are analyzed. Construction of the model was detailed. An analytical approach to the phenomena was introduced and the effects of a merger was evaluated. Strategies that would soften the merge effects and the resulting structure were exposed.

In *Chapter 6* our n-agent game simulator explained. Computational details are covered.

2. TWO AGENT GAMES MODEL

To have an insight on the fundamentals of competitions, we studied a two agent game model with respect to the three agent model [8]. In two agent model, game is a series of micro competitions between two competitors. These competitors are randomly picked from a large number of players. The players compete against each other with a given winning probability. The probability assignments depend on the points of the players. At the end of each competition there is always one winner which means there is no withdraw condition. No decline rate is defined in this model. Player that loses a competition does not lose points while winner gets one point.

First we pick two agents from a total set of N players. Let's say x stands for the highest point, and y stands for the lowest point among the selected agents. The microscopic rules of a competition for two agent model is,

$$\begin{aligned}
&(x > y) \Rightarrow (p, q),\\
&(x = y) \Rightarrow (\frac{p+q}{2}, \frac{p+q}{2}) \Rightarrow (\frac{1}{2}, \frac{1}{2}).
\end{aligned} \tag{2.1}$$

First line of Equation (2.1) implies that for all x greater than y, the player with the score x will advance the competition with a probability of p and the player with the score y will advance the competition with a probability of q. The second line shows that if x is equal to y, both will advance the competition with the same probability which is 1/2. This means that in this model agents with the same score will be evaluated on the basis of equal likelihood. If $p = 1$, the player with x points always wins and if $p = 0$ the player with y points always wins [4]. By using this microscopic model, we can observe the changes in number of teams in a particular score range through the function below.

$$f_x = \{ \text{ fraction of the number of agents that have x points } \}$$

For f_x is a scaled function,

$$\sum_x f_x = 1. \tag{2.2}$$

As we mentioned before, players do not lose point when they lose in a competition. For this reason changes in an arbitrary f_x occur when a player with $x-1$ points wins a game and enters f_x or a player with x points wins a game and leaves f_x.

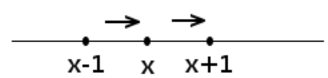

Figure 2.1. Entering and leaving f_x.

For a long time period, changes in f_x can be written as a derivative of time.

$$\frac{\partial f_x}{\partial \tau} = \sum_y f_{x-1} W_{x-1,y} f_y - \sum_y f_x W_{x,y} f_y \tag{2.3}$$

$W_{x,y}$ denotes the winning probability of the players. Its value was set by the microscopic rules in Equation (2.1), and it is the balancing factor of scaling in the Equation (2.3). On average, each player is expected to participate a single competition against all other players during a round, which defines the unit time and denoted by τ. Average point of the players is,

$$\overline{x}(\tau) \equiv \sum_{x=0}^{\infty} x f_x = \frac{\tau}{2}. \tag{2.4}$$

For an arbitrary x value, summing f_x would provide us a cumulative distribution for the rate of x,

$$F_x \equiv \sum_{x'=0}^{x-1} f_{x'} \tag{2.5}$$

which turns out that,

$$f_x = F_{x+1} - F_x \tag{2.6}$$

Then summing the Equation (2.3) over x would reveal the result below.

$$\frac{\partial F}{\partial \tau} = f_{-1} \sum_y W_{-1,y} f_y - f_{x-1} \sum_y W_{x-1,y} f_y \tag{2.7}$$

As there can not be a player with a negative point, $x = -1$ term is eliminated. Finally the equation becomes,

$$\frac{\partial F}{\partial \tau} = -f_{x-1} \sum_y W_{x-1,y} f_y. \tag{2.8}$$

If Equation (2.8) was expanded by using Equation (2.1) and Equation (2.3), we would get,

$$\begin{aligned}\frac{\partial F_x}{\partial \tau} =& - f_{x-1}[pF_{x-1} + q(1 - F_x)] \\ & - \frac{1}{2} f_{x-1}^2.\end{aligned} \tag{2.9}$$

The first line of the Equation (2.9) gives us the combination of interactions for the players with different scores which might be considered as bulk interactions [8]. The second line refers to the interactions where two players have the same scores. We should mention that players start the game in the same state. This refers that, in the very beginning of the game there would be many players with the same scores and because of this the second part of the equation is expected to affect the set of players. As time passes equalities will be broken and the bulk interactions will dominate the game. Therefore, in a continuum limit where the differences are expanded in terms of derivatives, the second part of the equation will be negligible and one may consider only bulk terms. As a result we will have,

$$\frac{\partial F_x}{\partial \tau} = -\frac{\partial F_x}{\partial x} G'(F) \tag{2.10}$$

where,

$$G'(F) \equiv pF + q(1 - F). \tag{2.11}$$

To analyze the winning rate for long time span [2] we normalize the points of the players by time. This gives us a chance to introduce a scaling solution to Equation (2.10) in an ansatz form. Let x denote the points of an arbitrary player, and τ unit time. By defining F as,

$$F(x, \tau) \Rightarrow F(z \equiv \frac{x}{\tau}) \tag{2.12}$$

we would get,

$$\frac{dF}{dz}[-z + G'(F)] = 0. \tag{2.13}$$

In this scaling equation F would be constant or $G'(F) = z$ [8]. A general form of the solution would be,

$$F(x, \tau) = \begin{cases} 0 & z < z_l \\ \Phi(z) & z_l \leq z \leq z_r \\ 1 & z \geq z_r \end{cases} . \tag{2.14}$$

We form a characteristic equation where $G'(F)$ is the speed of the curve emerging from x_0.

$$x(\tau) = x_0 + \tau G'(F(x_0, 0)) \tag{2.15}$$

The speed characteristics for a two player game are $G'(0) = q$, which represents the rate of the player with lowest points and it is denoted by (z_l), and $G'(1) = p$, which represents the rate of the player with higest points and it is denoted by (z_r). There are two regimes in two agent games. First is the case where $p > q$, and the second is the case where $q \geq p$. Figure 2.2 represents a graph solution for $p > q$. For this case, we have $F(z) = 1$ where $z > p$ and $F(z) = 0$ where $z < q$. From the ansatz,

$$G'(\Phi) = \frac{x}{\tau} \equiv z. \tag{2.16}$$

We get $\Phi(z)$ for the region $q \leq z \leq p$,

$$\Phi(z) = \frac{z - q}{p - q}. \tag{2.17}$$

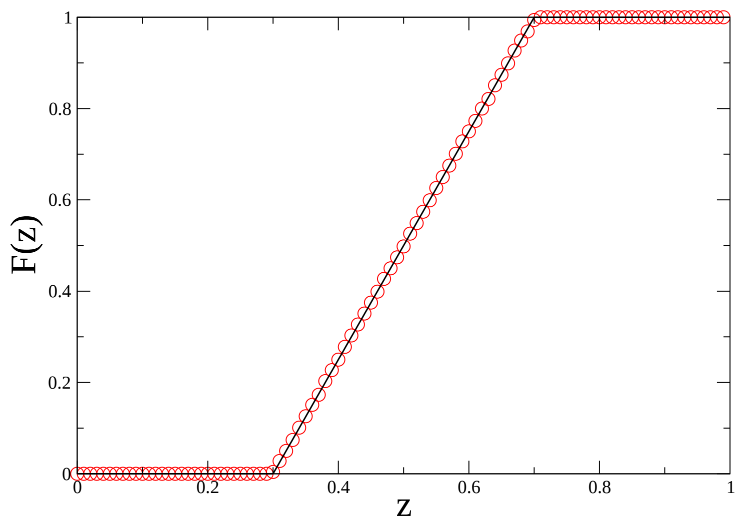

Figure 2.2. A graph for solution of $p > q$. The circles represent the simulation and the line represents the analytical solution.

For the case of $p \leq q$, the characteristics intersect. Players with higher points get lower winning probabilities against players with lower points which will lead to a shock solution in a hydrodynamical limit. The shock speed for any $G(F)$ is given by the equal area construction [8],

$$v = \frac{G(F_l) - G(F_r)}{F_l - F_r} \tag{2.18}$$

where F_l and F_r are right and left discontinuities. In this situation where $p < q$, $F_l = 0$and $F_r = 1$. The integral of $G'(F)$ is,

$$G(F) = \frac{(p - q)F^2 + qF}{2}. \tag{2.19}$$

When we solve Equation (2.18), we find $v = 1/2$. The shock will be at 0.5 and areas crossing the intersection point are same. Figure 2.3 represents a graph for the situation $p \leq q$.

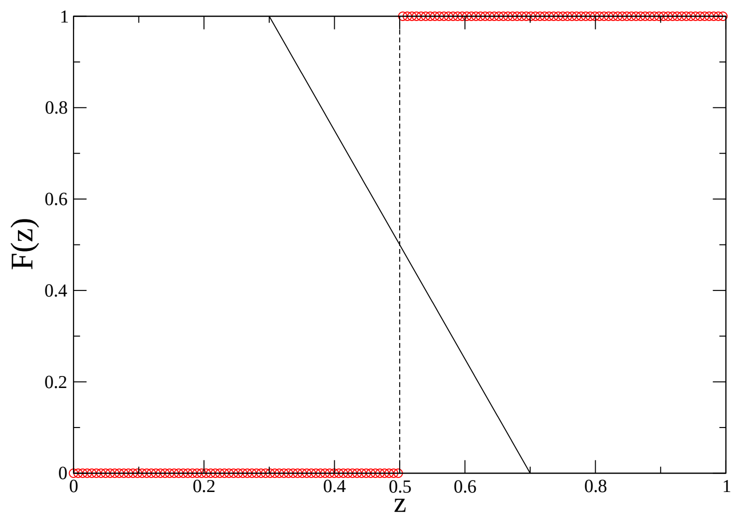

Figure 2.3. A graph for solution of $p \leq q$. The circles represent the simulation. The shock is at 1/2.

3. THREE AGENT GAMES MODEL

As we mentioned before this model was previously studied in detail by Mungan and Rador, 2007 [8]. The concept of the game is the same with the two agent model we studied in the previous chapter. The game is played among a large number of players. In this model we have three participating agents.

3.1. The Model

Points of the three agents, ordered from highest to lowest are denoted by x, y, z and winning probabilities are denoted by p, t and q respectively. As we mentioned earlier, agents participating the competition with the same score will be evaluated on the basis of equal likelihood. Accordingly our microscopic rules for this model would be,

$$\begin{aligned}
(x > y > z) &\Longrightarrow (p, t, q) \\
(x = y > z) &\Longrightarrow (\frac{p+t}{2}, \frac{p+t}{2}, q) \\
(x > y = z) &\Longrightarrow (p, \frac{t+q}{2}, \frac{t+q}{2}) \\
(x = y = z) &\Longrightarrow (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}).
\end{aligned} \tag{3.1}$$

As in two player games, we need the function f_x to observe the changes in the number of players in a particular score range. This function is a fraction of the number of agents over the whole game players.

$$\sum_x f_x = 1 \tag{3.2}$$

Changes in an arbitrary f_x occur when a player with $x-1$ points wins a game and enters, or a player with x points wins a game and leaves f_x. Therefore the changes in f_x with respect to time would be written as,

$$\frac{\partial f_x}{\partial \tau} = \sum_{y,z} f_{x-1} W_{x-1,y,z} f_y f_z - \sum_{y,z} f_x W_{x,y,z} f_y f_z. \tag{3.3}$$

Remember that $W_{x,y,z}$ denotes the winning probability of the players. Its value is set by the microscopic rules in the Equation (3.1). On the ground that we have three players, the average point of players is,

$$\overline{x}(\tau) \equiv \sum_{x=0}^{\infty} x f_x = \frac{\tau}{3}. \tag{3.4}$$

We remember from the previous chapter that the cumulative distribution of the point rate for an arbitrary points of x is defined as,

$$F_x \equiv \sum_{x'=0}^{x-1} f_{x'} \tag{3.5}$$

which turns out,

$$F_{x+1} \equiv \sum_{x'=0}^{x} f_{x'} \qquad \Rightarrow \qquad f_x = F_{x+1} - F_x. \tag{3.6}$$

If we get the sum of Equation (3.3) over x we get,

$$\frac{\partial F}{\partial \tau} = -f_{x-1} \sum_{y,z} W_{x-1,y,z} f_y f_z. \tag{3.7}$$

Note that we ignored the f_{-1} part of the equation since there can not be negative points. The right handside of the equation denotes the winning probability of the player with x points against all other competitors. As we mentioned earlier that $W_{x,y,z}$ stands for the winning probability of the ordered agents against its competitors and it is determined by the ordering of the points revealed in Equation (3.1). Summing over the rate equations in (3.3) we get a closed equation for the cumulative distribution,

$$\begin{aligned}\frac{\partial F_x}{\partial \tau} = & - f_{x-1}[pF_{x-1}^2 + 2tF_{x-1}(1-F_x) + q(1-F_x)^2] \\ & - 2\frac{(p+t)}{2} f_{x-1}^2 F_{x-1} \\ & - 2\frac{(t+q)}{2} f_{x-1}^2 (1-F_x) \\ & - \frac{1}{3} f_{x-1}^3 . \end{aligned} \tag{3.8}$$

The first line of Equation (3.8) represent the bulk of interactions between agents with different scores. The bulk part of the equation is quadratic since we have three agents. Second and third lines represent the case that two agents have the same points and the last line represent the case that every agent participating in the competition have the same score.

We can also show Equation (3.8) in a graphical representation that x-axis and y-axis refers to the possible opponents of the player with a score $x-1$. All players positioned on the lines according to their scores.

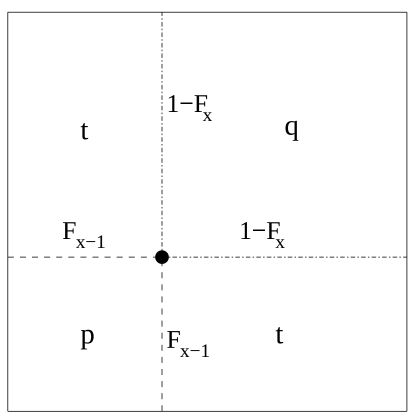

Figure 3.1. Graphical representation of the equation of cumulative distribution.

The player with $x-1$ points is represented with a filled circle. On Figure 3.1 dashed lines represent the opponents with less than $x-1$ points and the dot-dashed lines represent greater than $x-1$ points. The areas represent the bulk interactions where all players have different scores and the interfaces represent the cases where two players have the same score. The filled circle represents the last line of the equation where all players in the competition have the same score. From this figure one can calculate the bulk interactions by summing up the areas. A sum along the interfaces would give us the result of second and third lines of Equation (3.8). As we mentioned previously, the dot represents the cases where all players have the same points and in a continuum limit the dot and interface terms would break the balances between points aparting the points of the players from each other. This would lead the importance of the interface terms to decline and the bulk interactions become the dominant factor of the game. As a result, the Equation (3.8) may be written as,

$$\frac{\partial F_x}{\partial \tau} = -\frac{\partial F_x}{\partial x} G'(F) \tag{3.9}$$

with,

$$G'(F) \equiv pF^2 + t(1-F)F + q(1-F)^2. \tag{3.10}$$

In the Equation (3.10), G'(F) is not linear as it was in two agent games but it is a quadratic function. This means that G'(F) is not a function that monotonously increase or decrease and F has discontinuities like shocks. Defining the scaling ansatz,

$$F(x,\tau) \Rightarrow F(z \equiv \frac{x}{\tau}) \tag{3.11}$$

we would get,

$$\frac{dF}{dz}[-z + G'(F)] = 0. \tag{3.12}$$

When $G'(F) = z$, F is concave up or down in z.

3.2. The Regimes of Three Agent Games

The regimes emerged from the solution of F are:

- $C^- : p > t \geq q$ and $t < 1/3$,
- $C^0 : t = 1/3 > q$,
- $C^+ : t > 1/3$ and $p \geq t$,

- $C_S^+ : p < t$ and $q \leq 1/3$,
- $S : q \geq 1/3 > p$,
- $C_S^- : q > t$ and $p > 1/3$.

A general form of the solution for all regions is,

$$F(z) = \begin{cases} 0 & z < z_l \\ \Phi(z) & z_l \leq z \leq z_r \\ 1 & z \geq z_r. \end{cases} \tag{3.13}$$

As $G'(F) = z$, we find the roots of $\Phi_\pm(z)$ as,

$$\Phi_\pm(z) = \frac{q - t \pm [(q-t)^2 + (1-3t)(z-q)]^{1/2}}{1-3t}. \tag{3.14}$$

In the extreme competitive limit where p goes to one, Equation (3.14) would yield,

$$\Phi_\pm(z) = \sqrt{z}. \tag{3.15}$$

We introduce two metrics, namely “social indices ” [8], which define the score distributions of the players in their societies. Equation (3.16) is the ratio of the lowest and highest points and Equation (3.17) is the length of the score region that the agents are distributed over total points.

$$\omega \equiv \frac{f(z_l)}{f(z_r)} = \frac{F'(z_l)}{F'(z_r)} \tag{3.16}$$

$$\sigma \equiv \frac{z_r - z_l}{1} \tag{3.17}$$

Note that F is the cumulative distribution of all players in a particular score range and cannot have multiple values. In regimes that G’(F) is not monotonous we see regions with multiple values because of a shock. Those cases are solved with an equal area construction.

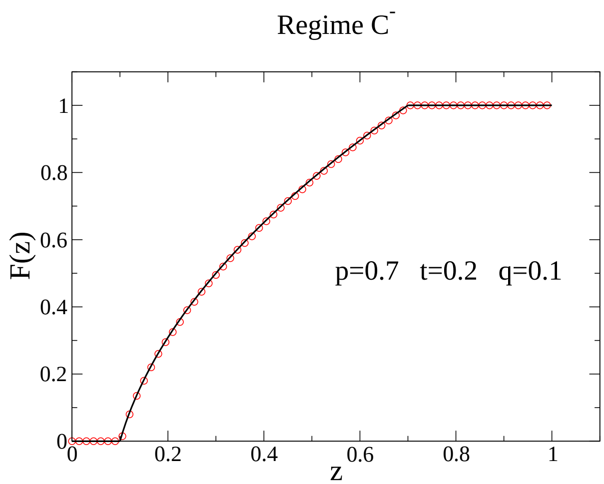

Figure 3.2. Graph for the solution in regime C^-. The circles represent the data from the simulation and line is the analytical solution.

3.2.1. C^- : Middle class society with mild hierarchy

In this regime $G'(F)$ is a monotonously increasing function from q to p. We can define $z_l = q$ and $z_r = p$. $\Phi_+(q) = 0$ and $\Phi_+(p) = 1$. From the Figure 3.2, we can see that F is a concave function which implies that most of the agents are in the lower range. This is also consistent with the fact that the players which are in the middle of the sorted agents win a competition with a probability less than 1/3. Observing the social indices we would get,

$$\omega_{C^-} = \sqrt{1 + \frac{(1-3t)(p-q)}{(q-t)^2}} > 1 \tag{3.18}$$

$$\sigma_{C^-} = p - q \tag{3.19}$$

where $\omega_{C^-} > 1$ shows the bias in the distribution that opposes two agent games. At an extreme point $p = 1 - 2t$ and $q = t$, Φ becomes,

$$\Phi(z) = \left[\frac{z-t}{1-3t}\right]^{1/2} \tag{3.20}$$

where ω diverges as a simple pole meaning that most of the agents are near the lowest scores in the situation that $z = q$.

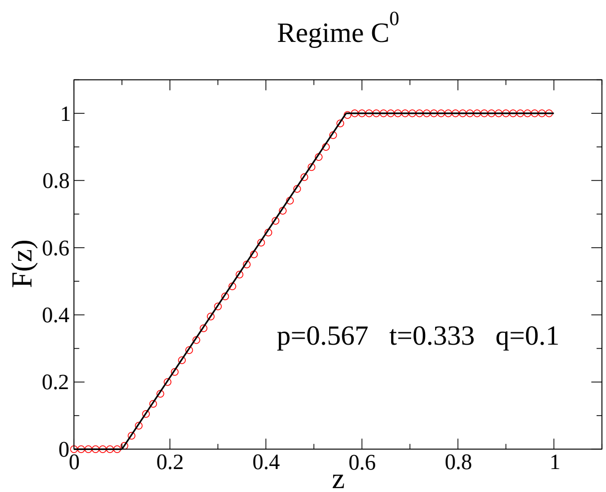

Figure 3.3. Graph for the solution in regime C^0. The circles represent the data from the simulation and line is the analytical solution.

3.2.2. C^0 : Pure middle class society

From the general form, we have $z_l = q$, $z_r = p$.

$$\Phi(z) = \frac{z - q}{p - q} \tag{3.21}$$

As middle player wins with a probability of 1/3, this solution is quantitatively same with the two agent games where $p > q$. The behavior of the middle player is like a random walk.

The social indices give us,

$$\omega_{C^0} = 1, \tag{3.22}$$

$$\sigma_{C^0} = p - q \tag{3.23}$$

which means the number of highest scored players are the same with the players with lower scores.

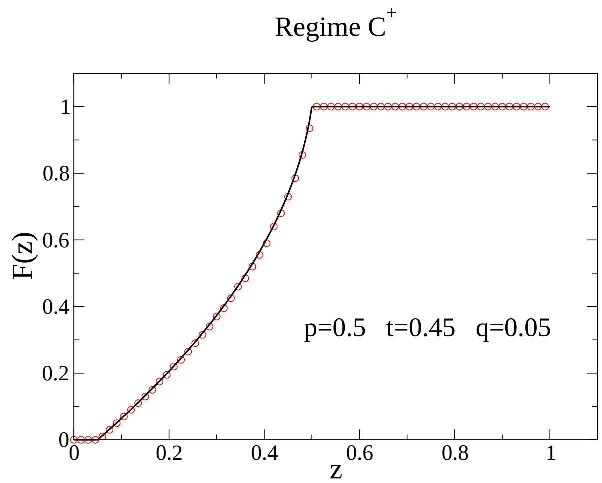

Figure 3.4. Graph for the solution in regime C^+. The circles represent the data from the simulation and line is the analytical solution.

3.2.3. C^+ : Middle class society with mild anti-hierarchy

From the general form of the equation we have $z_l = q$, $z_r = p$. $t > 1/3$ implies that the most of the players will be at the higher regions. An extreme case that $p = t$ and $q = 1 - 2t$ may be considered one step before a shock. For this case we have a particular $\Phi(z)$,

$$\Phi(z) = 1 - \left[1 + \frac{z-q}{1-3q}\right]^{1/2} \tag{3.24}$$

where the social indices are,

$$\omega_{C^+} = \sqrt{1 - \frac{|1-3t|(p-q)}{(q-t)^2}} < 1, \tag{3.25}$$

$$\sigma_{C^+} = p - q. \tag{3.26}$$

In this case $\omega_{C^+} = 0$ which confirms that most of the players are close to the highest score where $z = p$.

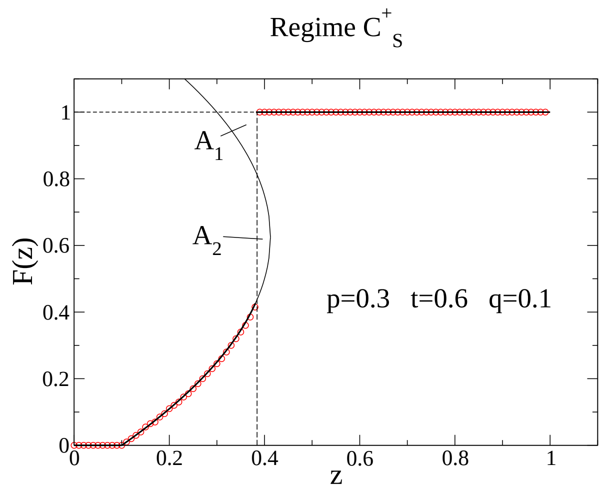

Figure 3.5. Graph for the solution in the regime C_S^+. The circles represent the data from the simulation and line represents the shock. Areas of A_1 and A_2 are equal.

3.2.4. C_S^+ : Anti-hierarchical society

In this regime $G'(F)$ is not monotonous which yields a shock front at z_r. From the general form of the solution, $z_l = q$ and the location of z_r is determined by equal area rule,

$$G'[\Phi(z_r)] = \frac{G(1) - G(\Phi(z_r)}{1 - \Phi(z_r)} \tag{3.27}$$

which yields a $\Phi(z)$ like,

$$\Phi(z_r) = \frac{1 - 3q}{2(3t - 1)} \tag{3.28}$$

and for $G'(\Phi(z_r))$ we obtain,

$$z_r = q + \frac{3q - 1}{4(1 - 3t)}(4t - q - 1). \tag{3.29}$$

There is a jump from $F = \Phi(z_r)$ to $F = 1$ and we have a shock discontinuity at z_r. Note that as $t > p$ in this game it is a disadvantage to have a higher score since middle points are favored in this society. This situation decelerates the rate of the players at the highest point level of the spectrum, which turns out that a player winning a game will be less favored and a player losing a game will be more favored having a chance to recover its position at the shock region.

Another interesting point is that when two players from the shock region and one player from the below region are selected, the winning probability of the higher two would be $(p + t)/2$ and the player with the lower point's probability will be q, where $q < 1/3$. This means that player with lower points would be disfavored automatically, resulting in a continuous population below the shock region. As we have discontinuity, in general form we have an F like below.

$$F(z) = \Phi(z) + [1 - \Phi(z_r)]\Phi(z - z_r) \tag{3.30}$$

where our social indices would become like,

$$\omega_{C_S^+} = \frac{constant}{\delta(0)}, \tag{3.31}$$

$$\sigma_{C_S^+} = \frac{4t - q - 1}{4}\left(\frac{1 - 3q}{3t - 1}\right). \tag{3.32}$$

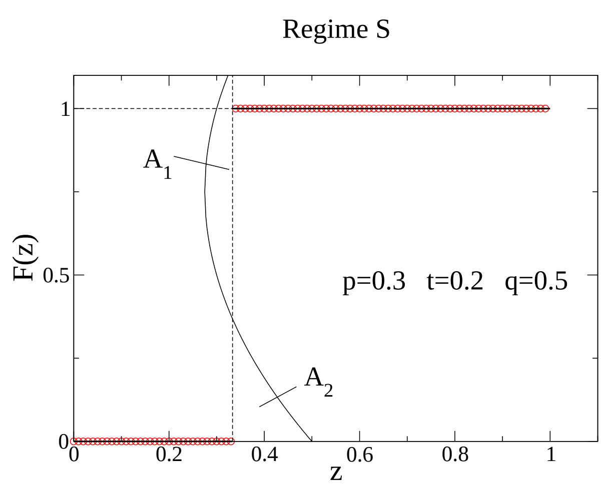

Figure 3.6. Graph for the solution in the regime S. The circles represent the data from the simulation and line represents the shock. Areas of A_1 and A_2 are equal.

3.2.5. S : Egalitarian society

In this regime $G'(F)$ is monotonously decreasing with F. Let's denote that the resulting discontinuity is z^*. The shock speed comes out as $1/3$, i.e. we have $z_l = z_r =$

$z^* = 1/3$. Note that in the case that $p > t$ and $p < 1/3$, we have a discontinuity at z covering whole interval of F so that the shock speed is again 1/3 and $z^* = 1/3$. The social indices are $\omega_s = 1$ and $\sigma_S = 0$ meaning that all players share the same wealth.

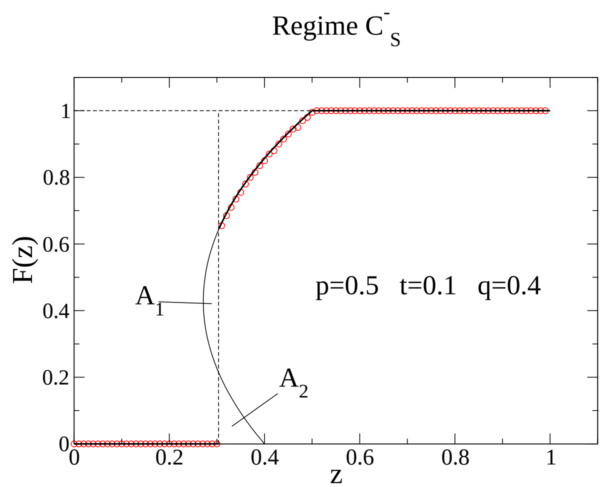

Figure 3.7. Graph for the solution in the regime C_S^-. The circles represent the data from the simulation and line represents the shock. Areas of A_1 and A_2 are equal.

3.2.6. C_S^- : Hierarchical society

In this regime shock is at z_l and $z_r = p$. From the equal area construction we get

$$\Phi(z_l) = \frac{3(q-t)}{2(1-3t)} \tag{3.33}$$

where z_l is found as,

$$z_l = q - \frac{3(q-t)^2}{4(1-3t)} \tag{3.34}$$

In this form, $q > t$ means that lower points are favored than middle points. And $p > 1/3$ yields that $p > (p+t)/2$. This means that players with higher points are trending towards the right side of the shock. The social indices are,

$$\omega_{C_S^-} = \frac{constant}{\delta(0)}, \tag{3.35}$$

$$\sigma_{C_S^-} = p - q + \frac{3(q-t)^2}{4(1-3t))}. \tag{3.36}$$

$\omega_{C_S^-}$ diverges much more strongly than the simple pole divergence in regime C^-.

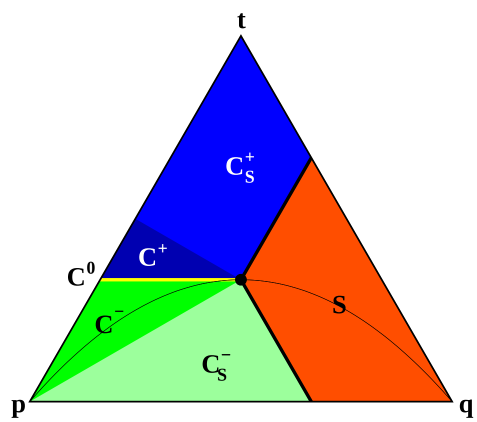

Figure 3.8. The phase diagram of the three-agent game presented on the plane $p + t + q = 1$. The thin black curve represent the resolution of a three-agent game in terms of two-agent mini tournament described in *chapter 5*.

Figure 3.8 is a phase diagram on the plane $p + t + q = 1$ that all regimes are represented in a combined form. The three corners represent the probabilities p, t, q equal to 1. For the edges of the triangle at the line $\overline{pt}$, $q = 0$ and at the line $\overline{qt}$, $p = 0$ and finally at $\overline{pq}$, $t = 0$. The dot gives the point where $p = t = q = 1/3$. And the thin curve represents the resolution of three agent games in terms of two agent games which is discussed in detail in *chapter 5*.

4. GENERALIZATION OF $n+1$ AGENT GAMES

The geometric representation in *chapter 3* for Equation (3.8) can be generalized to an n+1 agent game model. The concept of the game is same with two and three agent games. $n+1$ agents are selected from a large number of players. The distribution of the winning probabilities depend on the sorted points of the selected agents. The form of the equation is considered only in a hydrodynamical limit, where we focus only the bulk interactions. Game is constructed by picking a point in an n-cube and splitting its volume into n-1 planes by drawing line that passes through this point, each are orthogonal to each other and intersecting side. The binomial coefficient represents the number of equivalent n-volumes[8]. Accordingly, the formula comes out as,

$$\frac{\partial F}{\partial \tau} = -\frac{\partial F}{\partial x} G'(F). \tag{4.1}$$

where,

$$G'(F) = \sum_{k=0}^{n} \binom{n}{k} p_k (1-F)^k F^{n-k}. \tag{4.2}$$

As mentioned before, the winning probabilities are set by ordering the points of the players and kth highest score will get a probability of p_k.

$$\sum_{k=0}^{n} p_k = 1 \tag{4.3}$$

For $n+1$ players, the mean score is,

$$\overline{x}(\tau) = \frac{\tau}{n+1} \tag{4.4}$$

4.1. An Example of Analytically Solvable $n+1$ agent game

The generalized function is,

$$\frac{\partial F}{\partial \tau} = -\frac{\partial F}{\partial x} G'(F) \tag{4.5}$$

with,

$$G'(F) = \sum_{k=0}^{n} \binom{n}{k} p_k (1-F)^k F^{n-k}. \tag{4.6}$$

Let the p_k be derived by an s variable powered by k,

$$p_k = \alpha s^k, \tag{4.7}$$

forming a binomial expansion,

$$\alpha \underbrace{\sum_{k=0}^{n} \binom{n}{k} s^k (1-F)^k F^{n-k}}_{[s(1-F)+F]^n} \tag{4.8}$$

which makes the equation (4.5) analytically solvable. Sum of p_k written as a multiple of s^k gives us,

$$\sum_{k=0}^{n} p_k = 1 \Leftrightarrow \sum_{k=0}^{n} \alpha s^k = 1 \tag{4.9}$$

gives us a geometric series resulting with an analytical solution,

$$\sum_{k=0}^{n} \alpha s^k = \alpha \frac{1-s^{n+1}}{1-s} = 1. \tag{4.10}$$

It turns out that

$$\alpha = \frac{1-s}{1-s^{n+1}}. \tag{4.11}$$

Construction of α as a scaling variable accepts $p_k < 1$ for all values of s. From the ansatz,

$$F(z) = [-z + G'(F)] \tag{4.12}$$

we find,

$$G'(F) = \alpha[s + (1-s)F]^n = z \tag{4.13}$$

If we solve F from this argument, we get,

$$F(z) = \frac{\left(\frac{z}{\alpha}\right)^{\frac{1}{n}} - s}{1-s} \tag{4.14}$$

with α as in Equation (4.11).

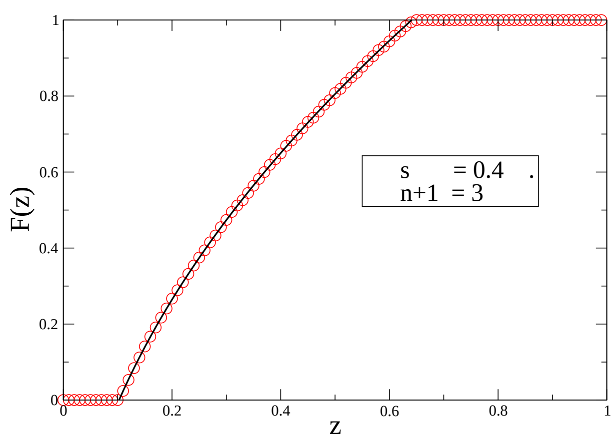

Figure 4.1. Graph solution of n+1 agent games where $n+1=3$ and $s=0.4$. The circles represent the simulation and the solid line represents the analytical solution.

5. MERGER DYNAMICS IN THREE AGENT GAMES

In previous chapters we observed the dynamics of multi-agent games in two agent, three agent and a generalized $n+1$ agent game models. All these models have common characteristics that a finite number of players are picked from a large group, the selected players are ordered with respect to their points from highest to lowest and they compete against each other with given winning probabilities according to their points. In all these games there is one probability set assigned among the competitors.

In merger model we studied a case where there are two different competition types with different probability sets. We studied this case in a condition that these competition types conflict with each other. To realize this idea, we chose competitive game as the first type of competitions where highest score is favored. This types of game is denoted as *"no-merger games"*. Second type has been merger game where middle score is favored.

In merger games weaker players combine their points and act as one strong player against the strongest only if it is an advantageous situation. It is not possible to study this case in a two agent model as we need at least two players to merger against a third one. This needs the case to be studied in three agent model at least.

In the case we observe, two players with lower points merge against the player with the highest point and because of the fact that there will be one winner in a competition, winning against the strongest player would lead the merging players play a new game against each other. This suggests a two agent game and it would be meaningful to resolve the game as a subset of the three agent game in terms of two agent mini tournaments.

5.1. Resolution of Three Agent Games in terms of Two Agent Games

Let the players participate in a competition be denoted by x, y, z, where x represent the player with highest point, y is the middle and z is the one with smallest point. Winning probabilities are represented by (p, t, q), respectively. Resolving the game as a subset of three agent games in terms of two agent games would mean that each player will play a two agent game against all its opponents. We introduce a single probability variable, (θ), denoting the winning probability of the player with higher point between the two competitors in a round of the tournament. Accordingly, the player with lower point will have $(1-\theta)$ probability of winning.

As there will only be one player that will increase its points in the competition, the winner of the tournament will be determined by counting the wins of players for every round of the tournament. If a player wins more rounds than its competitors, it increases its points. An analysis on the tournament gives us the following equations to determine the winning probabilities of the three players:

$$\begin{aligned} p &= \theta^2 + \frac{1}{3}\theta(1-\theta) \\ t &= \theta(1-\theta) + \frac{1}{3}\theta(1-\theta) \\ q &= (1-\theta)^2 + \frac{1}{3}\theta(1-\theta) \end{aligned} \tag{5.1}$$

The thin line in Figure 3.8 represents this kind of resolution. For $\theta > 1/2$, game results in in the C^- regime and the probability $\theta = 1/2$ means that $p = q = t = 1/3$, suggesting a shock and it is represented by a thick dot in Figure 3.8. For the situation $\theta < 1/2$, the shock remains as the game results in the S regime.

5.2. The Model

In the beginning of *chapter 5*, we mentioned that there are two types of competitions with different probability sets in merger model. One is a competitive game, namely no-merger game, and the second is merger game. While competitive games are played in the form of two agent mini tournaments, in merger games two players combine their points against the third one and act as a single player. Table 5.1 shows all possible merging combinations and the winning probabilities of the sorted agents for each case. x, y, z denote the points of the players ordered from highest to lowest while p, t, q represent the assigned probabilities respectively and θ denotes the winning probability of the player with highest score in a round of two agent mini tournament.

Table 5.1. Table of Merging Conditions

	no-merger	x-y	y-$z(y+z > x)$	y-$z(y+z < x)$	x-z
p =	$\theta^2 + \frac{\theta(1-\theta)}{3}$	$\theta^2 \downarrow$	$(1-\theta)$	θ	$\theta^2 \downarrow$
t =	$\frac{4\theta(1-\theta)}{3}$	$\theta(1-\theta) \downarrow$	$\theta^2 \uparrow$	$(1-\theta)\theta \downarrow$	$(1-\theta)$
q =	$(1-\theta)^2 + \frac{\theta(1-\theta)}{3}$	$(1-\theta)$	$\theta(1-\theta) \uparrow$	$(1-\theta)^2 \downarrow$	$\theta(1-\theta)$

In a merger game, it is expected that each merging player increase its winning probability compared to the probability it would get in a no-merger game. From Table 5.1, we can see that this situation occur only at the fourth column where the players with middle and the smallest points merge under the condition that the sum of their points is higher than the player with the highest point. We can define that a merger occur when,

$$\begin{aligned} &y + z > x, \\ &\bar{t} > t, \\ &\bar{q} > q, \end{aligned} \tag{5.2}$$

where $\overline{p}, \overline{t}, \overline{q}$ denote the winning probabilities of ordered players for merger game.

5.3. Exact Equation for Merger Model

The general formula of the interactions for merger model is,

$$\frac{\partial f_x}{\partial \tau} = -\frac{\partial}{\partial x}\left\{ f_x \int dydz f_y f_z W_{x,y,z} \right\}, \tag{5.3}$$

$$\frac{\partial F_x}{\partial \tau} = -\frac{\partial F_x}{\partial x}\left\{\int dydz f_y f_z W_{x,y,z}\right\}. \tag{5.4}$$

We remember from the previous chapters that, f_x is the fraction of the players with x points and F_x represents the cumulative distribution for the rate of x and $W_{x,y,z}$ denotes the winning probability which is assigned to the players with respect to the ordering of the selected players and the type of the game.

It is interesting that with these conditions a player with very small value of z points and a player with y points very close to x, can merge against x. This means that whole game is expected to be effected by merger. But an arbitrary player with r points as a middle player cannot merge with players lower than its score against a player with $2r$ or higher points, because there is no possibility that the sum of their points exceed $2r$. This is also true where r is the highest score for players with $r/2$ points. This suggestion gives us the boundaries for the interactions in the model. Let's denote p, t, q as the winning probabilities for a no-merger game, and $\overline{p}, \overline{t}, \overline{q}$ as the winning probabilities for a merger game. The interactions in merger model gives us the differential equation for Equation (5.4) as,

$$\frac{\partial F_x}{\partial \tau} = -\frac{\partial F_x}{\partial x}H \tag{5.5}$$

$$\begin{aligned}
H &= \overline{p}[F^2(x) - F^2(x/2)] + pF^2(x/2) + 2(p-\overline{p})\int_{x/2}^{x} dy\frac{\partial F}{\partial y}F(x-y) \\
&+ 2\overline{t}[F(2x) - F(x)]F(x) + 2tF(x)[1-F(2x)] + 2(t-\overline{t})\int_{x}^{2x} dy\frac{\partial F}{\partial y}F(y-x) \\
&+ \overline{q}[1 + F^2(x) - 2F(x)F(2x)] - 2qF(x)[1-F(2x)] + 2(q-\overline{q})\int_{2x}^{\infty} dy\frac{\partial F}{\partial y}F(y-x).
\end{aligned} \tag{5.6}$$

In Equation (5.6), x denotes the observed player and y denotes its opponent in a single microscopic competition of the two agent resolution of the game. Probability assignments are dependent not only the order of the points but also the merging option regulations. We can find some solutions in the extreme ranges where players will play either a merger game or a no-merger game by definition. But in the sections between these extreme ranges the interactions are non-deterministic and it is difficult to find an exact approach to Equation (5.6). By the help of numerical analysis we are able to find some explanation about merger model.

5.4. Numerical Approach for Merger Model

From the definitions in Table 5.1 and Equation 5.2, comparing no-merger game versus merger game yields that merger is effective at,

$$\theta > \frac{3}{5}. \tag{5.7}$$

Observing the results of games in the range of θ from Equation 5.7 to one, gives us an idea about the dynamics of merger. It gets clearer what happens on the regime at some extreme points. Figure 5.1 represents a graph solution where θ is close to 3/5. The solid line in the graph shows a competitive game which results in C^-. The circles represent the simulation of merger game which results in a shock solution. This yields an anti-hierarchical society which is denoted by C_S^+ for three agent games. z_l represents the leftmost point of the shock and z_r is the rightmost of the shock.

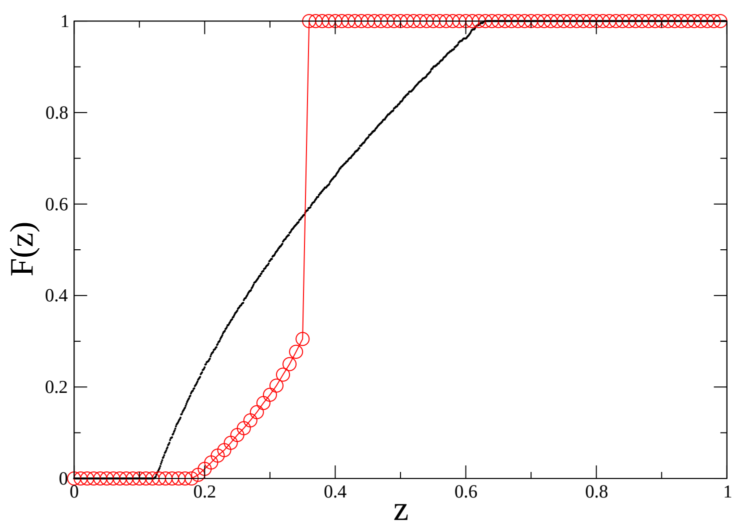

Figure 5.1. Graph for the solution where $\theta = 0.75$, $z_r < 2z_l$. The circles represent the simulation and solid line represents the game without a merger option

In Figure 5.1 $z_r < 2z_l$ suggests that when we pick two arbitrary players, it is certain that sum of their points will always exceed the maximum points gained.

This yields a game which is totaly dominated by merger. At an extreme point where $z_r = 2z_l$, the players in the z_r bunch start to play no-merger game against the players near z_l. Calculating θ for $z_r = 2z_l$ by the help of Equation (3.29) yields,

$$\theta = \frac{\sqrt{13}+1}{6} \approx 0.77. \tag{5.8}$$

For θ values that are higher than Equation (5.8), $z_r > 2z_l$ holds. The effect of the no-merger games show it self as a collapse in the leftmost part of the graph. This forms another shock with boundaries z_l and say z^*. The players in z_r play no-merger games with the players lower than z^*. Figure 5.2 shows that situation.

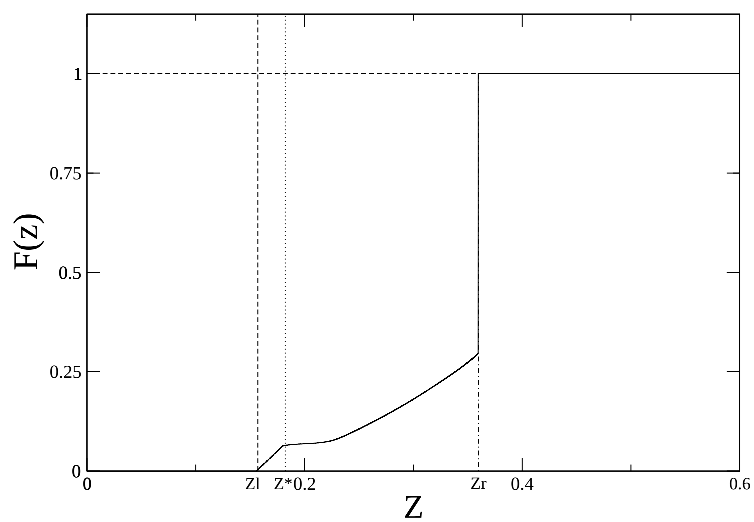

Figure 5.2. Graph for the solution where for $\theta = 0.79$, $z_r > 2z_l$. The solid line represents the simulation.

At the extreme point of $z_r = 2z^*$, the players at z_r become completely isolated from mergers against the players at z^* but there are still players between z^* and z_r and these players will still play merger games against the players at z^*. $z_r = 2z^*$ occur at $\theta \approx 0.8$. We should mention that this value of θ is observed by numerical analysis and we do not have an analytical explanation for this case.

A slight increase of θ from 0.8 leads the players between z^* and z_r deplete the area. This forms a complete isolation of the players at z_r from merger games against all lower bunches. Figure 5.3 represents this effect. It is not necessary that the players in z_r have the same points. Most of the players in this bunch have points very close to each other. This shows that the players in z_r will mostly play merger games with the players in its own range which will be mentioned as *"self games"*.

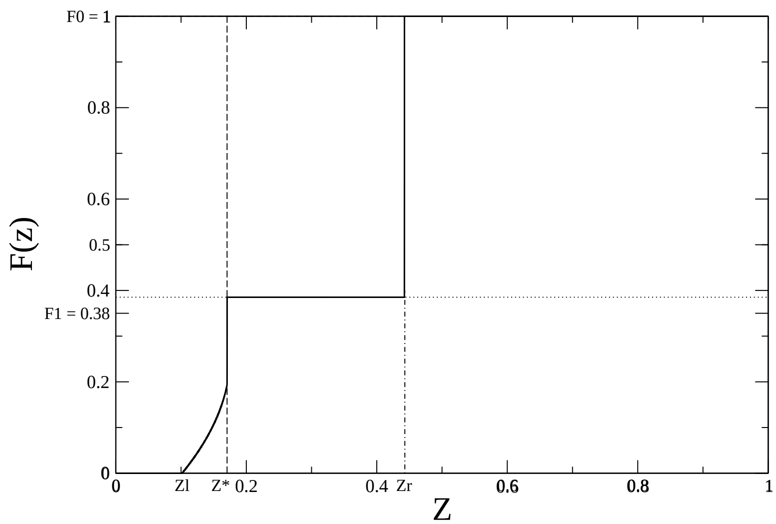

Figure 5.3. Simulation results for $\theta = 0.85$, $z_r > 2z^*$.

The leftmost part of the Figure 5.3 is merger dominated in itself. It has a similar structure with Figure 5.2. Denoting L as a player from left bunch and R as a player from right bunch, a list of interactions between two bunches for every possible combination that three players are picked in Figure 5.3 would be:

- $L - L - R$: as $z_r > 2z^*$ all the players in the right bunch are more than two times greater than that of left bunch, leadingto no-merger game for all competitions in this combination.
- $L - R - R$: because of the fact that point differences of $R - R$ are overcome by that of L, this combination ends up almost always a merger game.
- $L - L - L$: this is a merger dominated self game.
- $R - R - R$: this is also a self game in the boundary of z_r with little differences in players points and they will almost always play merger game.

Increasing θ generates a result that $z^* \geq 2z_l$. In this case z^* also protects itself from merger against z_l and becomes a separate bunch. This is determined as a self similar structure. The graph at very high values of θ, showing the isolated bunches and self similar structure, is represented in Figure 5.4.

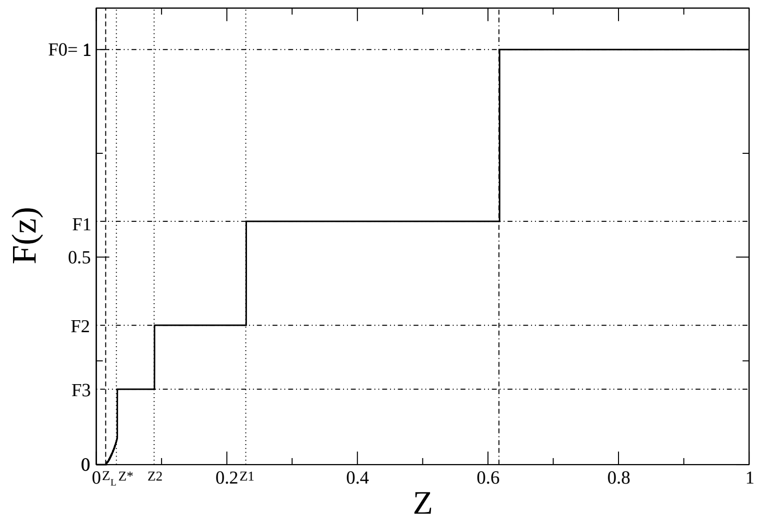

Figure 5.4. Graphical representation of the simulation for $\theta = 0.97$.

Figure 5.5 represents the players at the leftmost point of an $F(z) - z$ graph for all effective θ values. From the derivative of Figure 5.5 with respect to θ, we can see the transitions of the shocks.

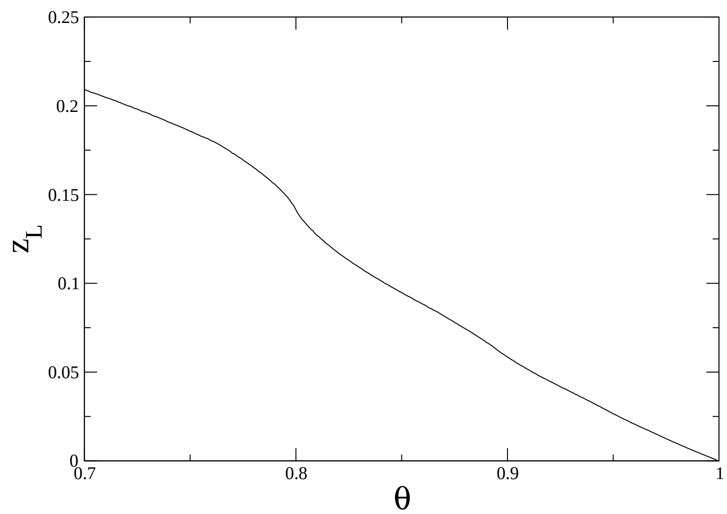

Figure 5.5. The leftmost agents for all θ values.

Figure 5.6 shows the two type transitions for the shocks. First type represents the cases where $z_r = 2z*$ and become separated from the left bunch. The second transition is the case where $z^* = 2z_l$ which starts a new shock. From the Figure 5.6, we can see the self similar structure of the game.

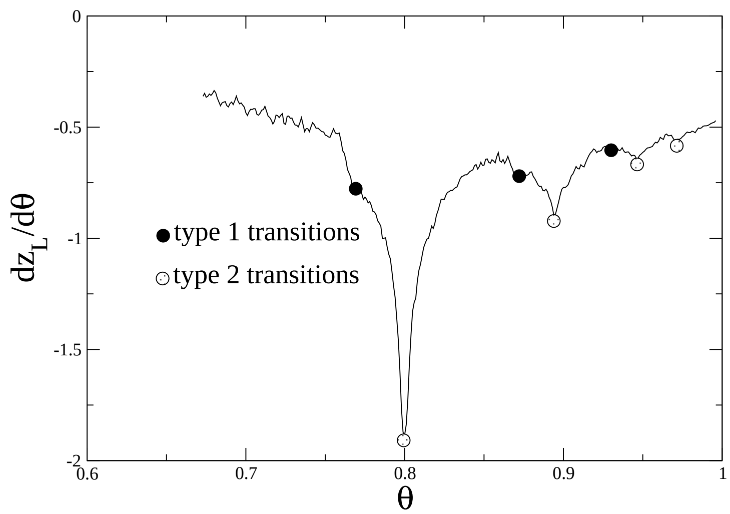

Figure 5.6. The transitions of the shocks across the leftmost bunch

Observing the interactions between bunches we form an ansatz,

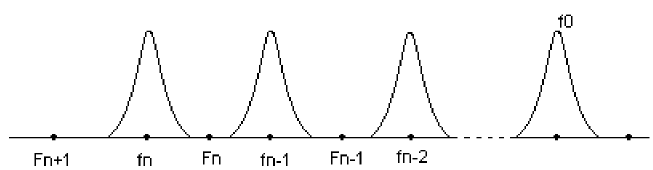

Figure 5.7. The interactions between separate bunches.

In Figure 5.7, f_n denotes the players in the bunch and F_{n+1} denotes the players below f_n. We assume that bunches are separate enough so that a given bunch, f_n, is protected against mergers of any two players from the bunches below. Note that, as bunches emerge in the direction from one to zero, we enumerate the bunches from the right to the left. So iterating the possible combinations we listed for Figure 5.3, we have,

$$\frac{\partial F_n^x}{\partial \tau} = -\frac{\partial F_n^x}{\partial x} H \tag{5.9}$$

where

$$
\begin{aligned}
H = & pF_{n+1}^2 \\
& + 2F_{n+1}\{\overline{p}(F_n^x - F_{n+1}) + \overline{t}(F_n - F_n^x)\} \\
& + 2tF_{n+1}(1 - F_n) \\
& + 2(1 - F_n)\{t(F_n^x - F_{n+1}) + q(F_n - F_n^x)\} \\
& + (1 - F_n)^2 q \\
& + (\overline{q} - q) \sum_{k=n-1}^{0} (F_k - F_{k+1})^2 \\
& + \overline{p}(F_n^x - F_{n+1})^2 + 2\overline{t}(F_n^x - F_{n+1})(F_n - F_n^x) + \overline{q}(F_n - F_n^x)^2
\end{aligned}
\tag{5.10}
$$

For a player with x points in the n^{th} bunch, the last line of H represents the self game terms which means all players are in f_n with the possibilities that the opponents are lower or higher than x. The first line represents the case that both opponents are from the below range playing a no-merger game. The second line holds the case that an opponent in the same bunch or x against an opponent from the same bunch merges with a player from the lower range. Third line is a no-merger game with one lower and one upper bunches. While the fourth line represents a case where one opponent is from the upper bunch and the other opponent is in f_n with either lower or higher points than x. In the fifth line all opponents of x are from the upper bunches. The sixth line represents the interactions for the regions between bunches. By borrowing F_k terms of Equation (5.10), we find the characteristic polynomials.

From this sense, in Figure 5.8 by borrowing the location of F_1, we determine the values of z_r, $z*$ and z_l. Because of the equal area rule for the shocks, the areas across the curves and the graph are equal and the area under the solid line in Figure 5.8 is 2/3 by construction. For more than two shocks as in Figure 5.4, borrowing F_k is not enough for a direct approach.

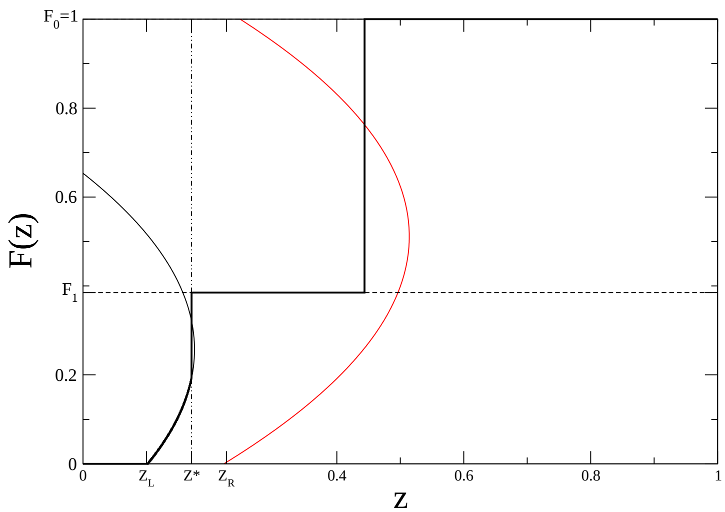

Figure 5.8. Simulation results for $\theta = 0.84$. Curves represent the characteristic polynomials.

At the extreme competitive limit where value of $\theta = 1$,

$$\begin{aligned} p, t, q &\Longrightarrow 1, 0, 0 \\ \overline{p}, \overline{t}, \overline{q} &\Longrightarrow 0, 1, 0 \end{aligned} \tag{5.11}$$

the Equation (5.9) becomes,

$$\begin{aligned} \frac{\partial F_n^x}{\partial \tau} = -\frac{\partial F_n^x}{\partial x}\big\{ & F_{n+1}^2 \\ & + 2F_{n+1}(F_n - F_n^x) \\ & + 2(F_n^x - F_{n+1})(F_n - F_n^x)\big\}. \end{aligned} \tag{5.12}$$

From our observations we expect that the shocks occur in a scaling behavior in the extreme competitive limit. To study the scaling let us isolate the first shock and consider the system as such. We assume the shock is protected from merger of the two players from the below range.

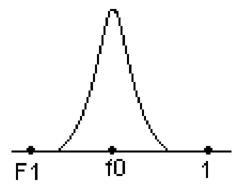

Figure 5.9. Represents the isolated rightmost shock region

We separate the total interactions into two parts. f_0 denotes the players in the bunch and f_L denotes the players below the bunch. Accordingly we have,

$$f^x = f_L^x + f_0^x \tag{5.13}$$

Observing these two parts separately we have two equations:

$$\frac{\partial f_0^x}{\partial \tau} = -\frac{\partial}{\partial x} f_0^x \int (f_L^y + f_0^y)(f_L^z + f_0^z) W_{x,y,z} \tag{5.14}$$

$$\frac{\partial f_L^x}{\partial \tau} = -\frac{\partial}{\partial x} f_L^x \int (f_L^y + f_0^y)(f_L^z + f_0^z) W_{x,y,z} \tag{5.15}$$

x, y, z denotes the selected players. $W_{x,y,z}$ denotes the winning probability for the players. Expanding Equation (5.14) will give all interactions for a selected player from the first shock.

$$f_0^x \int f_L^y f_L^z W x, y, z \tag{5.16}$$

Equation (5.16) means that two players are selected from left bunch and so, it will be a no-merger game.

$$f_0^x \int f_0^y f_L^z W x, y, z \tag{5.17}$$

Equation (5.17) refers that players with lower scores one from inside the bunch and the other is from the below part of the bunch. Which emerges almost always as a merger game.

$$f_0^x \int f_0^y f_0^z W x, y, z \tag{5.18}$$

In Equation (5.18), all players are from inside the bunch. As the sum of the points of lower scored players will always be higher than the higher one, this part is merger dominated. So the formula from the Equation (5.14) comes out like this,

$$\begin{aligned} \frac{\partial F_0^x}{\partial \tau} = &- \frac{\partial F_0^x}{\partial x}\{pF_1^2 + 2\overline{p}F_1(F_L^x - F_1) \\ &+ 2\overline{t}(F_1)(1 - F_0^x) + \overline{p}(F_0^x - F_1)^2 \\ &+ 2\overline{t}(F_0^x - F_1)(1 - F_0^x) + \overline{q}(1 - F_0^x)^2\} \end{aligned} \tag{5.19}$$

For the Equation (5.15) the expansion reads the combinations as follows:

$$f_L^x \int f_L^y f_L^z W x, y, z, \tag{5.20}$$

Equation (5.20) reads that all players are in the left bunch, namely a self game of left part. As they are all within boundaries, it is a merger game.

$$f_L^x \int f_L^y f_0^z W x, y, z, \tag{5.21}$$

Equation (5.21) shows that two players will be from left and one player from f_0 bunch. As we mentioned before we assume that f_0 is protected from mergers of lower regions. Therefore, it is a no-merger game.

$$f_L^x \int f_0^y f_0^z W x, y, z, \tag{5.22}$$

In Equation (5.22), two players are in the f_0 bunch and their points are close to each other. The lower one of these two will always catch up the higher one by merging with any of the players from left. This case is also a merger game.

From these arguments we can form Equation (5.15) like,

$$\frac{\partial F_L^x}{\partial \tau} = -\frac{\partial F_L^x}{\partial x}\{\int f_L^y f_L^z W_{x,y,z} + 2(1-F_1)tF_L^x \\ + 2(1-F_1)q(F_1 - F_L^x) + (1-F_1)^2\overline{q}\}. \tag{5.23}$$

To observe the scaling behavior we define,

$$F_L^x = F_1 \Phi_L^x \tag{5.24}$$

where $\Phi_L^x \epsilon [0,1]$ since $F_L^x \epsilon [0, F_1]$. Using this definition we can form the Equation (5.23) scaled by F_1, which yields,

$$\frac{\partial \Phi_L^x}{\partial \tau} = -\frac{\partial \Phi_L^x}{\partial x}\{F_1^2 \int \frac{\partial \Phi_L^y}{\partial y}\frac{\partial \Phi_L^z}{\partial z} W_{x,y,z} \\ + 2t(1-F_1)F_1\Phi_L^x \\ + 2q(1-F_1)F_1(1-\Phi_L^x) \\ + \overline{q}(1-F_1)^2\}. \tag{5.25}$$

In the extreme competitive limit defined in Equation (5.11), all parts on the right hand side of the Equation (5.25) except the first line will vanish. When we look at the Equation (5.26) we will see the self similar structure of the game.

$$\frac{\partial \Phi_L^x}{\partial \tau} = -\frac{\partial \Phi_L^x}{\partial x}\left\{F_1^2 \int \frac{\partial \Phi_L^y}{\partial y}\frac{\partial \Phi_L^z}{\partial z} W_{x,y,z}\right\} \tag{5.26}$$

F_1^2 acts like a scaling factor of the original formula, which we introduced at equation (5.6). The scaling ansatz which we first defined in two agent games in Equation

(2.13) becomes,

$$\frac{d\Phi_L}{dz}\Big[-\frac{z}{F_1^2}+G'[\Phi_L]\Big]=0. \qquad \Big(z=\frac{x}{\tau}\Big) \tag{5.27}$$

We scaled F with a variable Φ and z with its square. To observe this scale in the whole regime we apply it to Equation (5.5). The shock locations are found with the equal area rule which yields,

$$v_n=\frac{1}{3}[F_n^2+F_nF_{n+1}+F_{n+1}^2] \tag{5.28}$$

In the view of the scaling ansatz we assume

$$F_n=\Phi^n \tag{5.29}$$

which yields,

$$f_n=(1-\Phi)\Phi^n \tag{5.30}$$

$$\sum_{n=0}^{\infty} f_n=1. \tag{5.31}$$

Forming Equation (5.28) with the help of Equation (5.29), we would get,

$$v_n=\Big(\frac{1+\Phi+\Phi^2}{3}\Big)\Phi^{2n} \tag{5.32}$$

where,

$$v_0=\Big(\frac{1+\Phi+\Phi^2}{3}\Big). \tag{5.33}$$

v_0 denotes the location of the first shock. The mean speed emerges from the construction of the game which is,

$$\sum_{n=0}^{\infty} v_n f_n = \frac{1}{3}. \tag{5.34}$$

Because of this we cannot use it to find the value of v_0 and we take v_0 by applying an extrapolation on the graph in Figure 5.10 which shows the rightmost points for all θ values. From this graph we find the value of v_0 as,

$$v_0 = \frac{2}{3} \tag{5.35}$$

This value means that the fastest moving agents move twice as fast as the mean speed of the whole players. Solving Equation (5.33) with Equation (5.35) we find,

$$\Phi = \frac{\sqrt{5}-1}{2} \qquad \Longrightarrow \qquad \text{(golden ratio -1)} \tag{5.36}$$

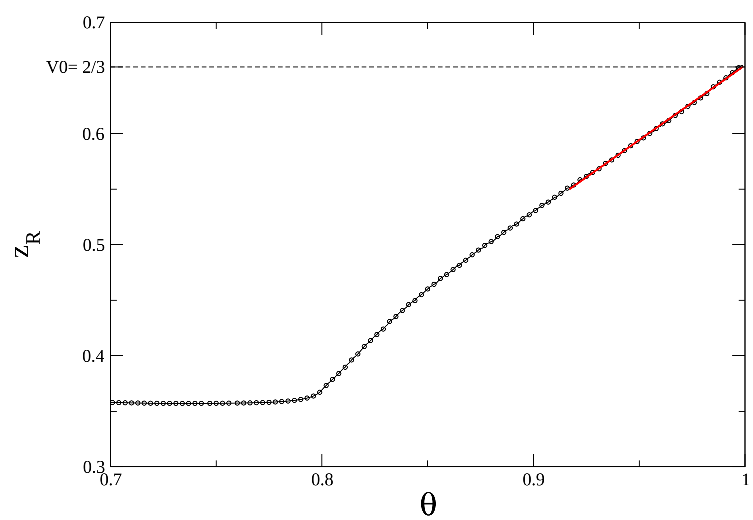

Figure 5.10. Represents the location of the rightmost shock. Red line represents the extrapolation on the data

In the extreme competitive limit where p and $\bar{t}$ go to one, and all other possibilities go to zero, one would ask that why the player with the lowest score participates a merge while the winning probability it would get from a merge, denotingly $\bar{q}$, goes to zero. That is because q, the probability it would get from a no-merger game, is also goes to zero. Observing this in a case that θ is very close to one, we assume that,

$$\theta = 1 - \epsilon. \tag{5.37}$$

where ϵ is a very small number. From the equations in Table 5.1 we would get,

$$\begin{aligned} q &= \epsilon^2 + \frac{\epsilon(1-\epsilon)}{3} \approx \frac{\epsilon}{3}, \\ \overline{q} &= \epsilon(1-\epsilon) \approx \epsilon. \end{aligned} \tag{5.38}$$

From the expressions in 5.38 we would understand that, by participating in a merge, the player with lowest score increases its winning probability approximately three times compared to the the winning probability it would get from a no-merger game. That's why the player with lowest score among the selected three players would participate a merge in extreme competitive limit even the probability it would get from a merge goes to zero.

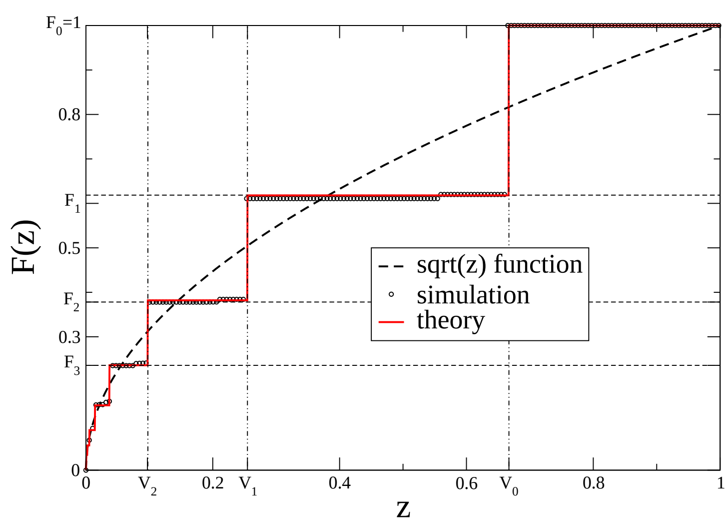

Figure 5.11. Simulation results for $\theta = 1$, representing the extremely competitive limit of merger game.

Figure (5.11) represents the graph solution of merger model in the extreme competitive limit. The circles represent the simulation output, the solid line represents the theoretical solution, the curve is the graph representation of $\sqrt{z}$ which is the solution of competitive games in extreme competitive level introduced by Equation (3.15). We can see from Figure 5.11 that merger model confirms the competitive pattern of the game. But the effect of merger games yields sub-societies in the regime. In an extremely competitive limit the emergence of these sub-societies show a scaling behavior. In competitive games where $p > t > q$, most of the players are in the lower ranges. In merger model, players are distributed in separate sub-societies and for an individual player moving from one society to another becomes very hard when these sub-societies become isolated from other sub-societies.

5.5. Restraining Merger

We observed that merger game significantly increases the competitiveness of the regime which yields sub-societies in a competitive game. We introduce an approach to constraint this effect.

As we mentioned before, on average, each player is expected to participate a single competition against all other players during a round, which defines the unit time and denoted by τ. Because of this the maximum points that a player may theoretically gain is equal to τ. As a restriction, we suggest that a merger is allowed if the points of all players participating in a competition are higher than $\tau/2$.

$$x > y > z > \tau/2 \tag{5.39}$$

We study this case separating the points into two parts .The first part is the case that all players have points with less than $\tau/2$ and For the second part, at least one player will be higher than $\tau/2$. L denotes the group of players with points that are equal to or lower than $\tau/2$ and R denotes the group of players with points that are higher than $\tau/2$. p, t, q denote the winning probabilites for a no-merger game and $\overline{p}$, $\overline{t}$, $\overline{q}$ denote the winning probabilities for a merger game.

The first part is definitely a no-merger game. All the players are in L and there is no possible combination for a merger. For this part, the interactions of the selected player according to their ordering yields,

$$H_L = pF^2 + 2t(1 - F)F + q(1 - F)^2. \tag{5.40}$$

We study the second part by observing the selected players individually. We know that there will be at least one player from R. Let's denote the points of this player as x. If all the opponents of the player with x points are from L, than there will be a no merger game. If one of the players are from L there may a merger or no merger game, and if all players are from the group R there will be merger game. This observation yields,

$$\begin{aligned} H_R =& pF^2 + (\overline{p} - p)(F - \overline{F}) + 2t\overline{F}(1 - F) && (5.41) \\ &+ 2\overline{t}(1 - F)(F - \overline{F}) + \overline{q}(1 - F)^2 && (5.42) \end{aligned}$$

$\overline{F}$ denotes the value of F at $z = 1/2$. We found that $H_L(\overline{F}) \neq H_R(\overline{F})$ where the difference is small. The points z_l and z_r are observed in the extreme competitive limit. In the extreme competitive limit, where θ, p and $\overline{t}$ goes to 1, the solution of F is,

$$F(z) = \begin{cases} \sqrt{z} & z < 1/2, \\ \frac{1}{2}[\sqrt{2} + 1 - \sqrt{2}\sqrt{1 - z}] & 1/2 \leq z \leq z_r, \\ 1 & z > z_r, \end{cases} \tag{5.43}$$

with,

$$\overline{F} = 1/\sqrt{2} \tag{5.44}$$

$$z_r = \frac{2\sqrt{z} - 1}{2} \tag{5.45}$$

In this limit ,

$$z_r - z_l = H_R(\overline{F}) - H_L(\overline{F}) = (\overline{q} - q)(1 - \overline{F})^2 \tag{5.46}$$

as we can write q and $\overline{q}$ in terms of θ we find,

$$\overline{q} - q = \theta(1-\theta)\frac{2}{3} - (1-\theta)^2. \tag{5.47}$$

$$\tag{5.48}$$

For θ goes to 1, we can define,

$$\theta = 1 - \epsilon \tag{5.49}$$

where ϵ has a very small value and equation (5.47) becomes,

$$\overline{q} - q = (1-\epsilon)\frac{2\epsilon}{3} - \epsilon^2 \tag{5.50}$$

$$\tag{5.51}$$

Therefore Equation (5.46) can be written as ,

$$z_r - z_l = \epsilon\left\{\frac{2}{3} - 5\epsilon 3\right\}(1-\overline{F})^2 \tag{5.52}$$

$(1-\overline{F})^2$ is also very small since $\overline{F}$ is not very small. So in general $z_r - z_l$ is very small and they are equal at $z = 1/2$.

Figure 5.12 represents the solution for the restricted merger game. We can see from the figure that left handside of the graph confirms the competitive games solution for the extreme competitive limit as only no-merger games were played in this region. The right handside shows a mild convergence due to the merger effect. In this configuration only the riches merge against each other. As we can see from the Figure 5.12 there are no sub-societies in this game. The seperation at $z = 1/2$ is very small for all values of θ and it vanishes at the extreme competitive limit.

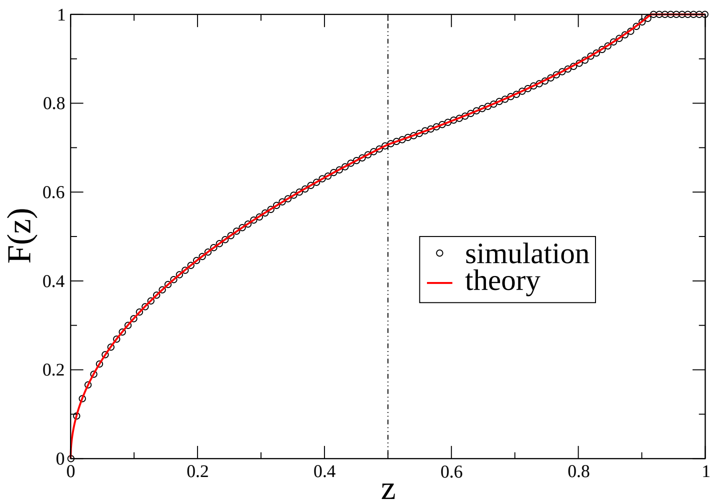

Figure 5.12. Represents the solution for restricted merger game in the extreme limit of competitiveness.

6. SIMULATION

We have four topics that we observed in this research,

- Two agent games
- Three agent games
- n+1 agent games
- Merging option in three agent games

In these simulations we observed the cumulative distribution of the players amongst their points normalized by time, ($F(z) - z$ graphs), to analyze the changes in social regimes. Games have generic rules which only differentiate by the characteristics of each model. Observing these circumstances we developed a single n-player simulation engine where we give all requirements as program arguments. Merger option is implemented and observed only for three agent games.

The arguments of the program cover the total number of players, (N), the number of players that participate in a single competition, (n), this is also one of the identifiers for the model we run the simulator, the time variable, ($TOUR$), histogram range, ($HIST$), merger condition. If merger strategy is chosen rather than no-merger game then θ also stated here. For the last set of the arguments the winning probability configuration for no-merger game is stated here, (p_k). Whe the program is executed, arguments are parsed and necessary arrays are allocated dynamically according to the arguments. We used *gsl_block* [1] data types for an accurate allocation and ease of use. We also used *gsl random number generator* with the default generator algorithm known as *Mersenne Twister generator* , introduced in 1998 by Makoto Matsumoto et.al. [9], which provides a period of about 10^{6000}.

[1]GSL is GNU Scientific Library. One can obtain the reference manual from http://www.gnu.org/software/gsl/

The game starts in the loop limited with $TOUR$ parameter where each loop consists of N/n competitions. This structure provides ease in the calculation of theoretical maximum points that a player would gain and one can expect that it provides statistically every player participating a competition with all other players in a hydrodynamic limit of time.

A competition starts with selection of (n) number of players . After the selection the points of the players are sorted from highest to lowest and their winning probabilities are set. Same points are be treated on the basis of equal likelihood.

If it is a merger game than the merging conditions are checked and the winning probabilities are recalculated which are covered in the previous chapter. A comparison on the merger probabilities and the no-merger probabilities provides us if it is a convenient condition to merge on the base of the rule (5.2). As we gather all the information we need to play a competition we initiate the competition by picking a uniform random number. The location of this number in the line of the winning probabilities of the players in a cumulative distribution determines the winner of the game where a player increases its points by one.

For a merger game, three agent games are simulated as a resolution of the game in two agent mini tournament. From the Table 5.1 we can see the probability distributions amongst the players according to the games characteristic as a merger or no-merger game is played. So with the simulation one would prefer either producing all that six regulatory variables with the help of a unique variable, θ, or can give an arbitrary no-merger probability set and observe an individual case. As an example, in the range of $\theta = 0.95$ observing the condition 1 where merging option is dependent to the rules

$$\bar{t} > t$$

$$\bar{q} > q$$

$$M + S > L$$

We can check from Table 5.1 that simulator will calculate the no-merger probabilities as, $p = 0.919$, $t = 0.063$, $q = 0.018$. The winning probabilities for merger games will be $\overline{p} = 0.05$, $\bar{t} = 0.9025$, $\overline{q} = 0.0475$. The resulting graph of this configuration is represented in Figure 6.1. This game is a merger dominated game with multiple shocks.

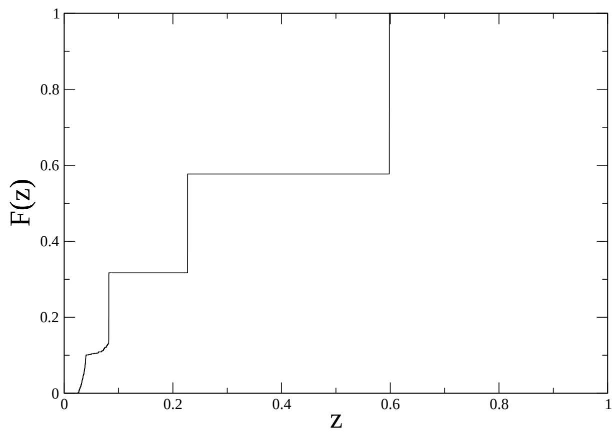

Figure 6.1. Simulation results for $\theta = 0.95$. Simulator calculated probabilities are,$p = 0.919$, $t = 0.063$, $q = 0.018$.

If one configures a game with the same θ and define no-merger probabilities as $p = 0.7, t = 0.2, q = 0.1$, the program will calculate the winning probabilities for merger games same with the example above. But one can see from Figure 6.2 that this configuration is not influenced by the merger option. It is because in this configuration players can merge only at interface terms where $y = z$. At bulk interactions $\overline{q} < q$ and players cannot merge.

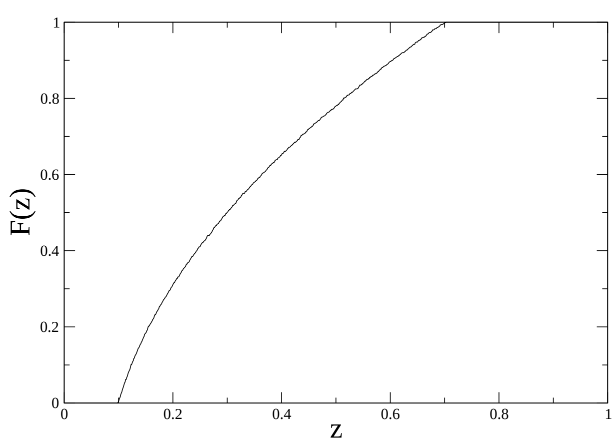

Figure 6.2. Simulation results for $\theta = 0.95$, $p = 0.7, t = 0.2, q = 0.1$.

Note that the process of the simulation has characteristics of Monte Carlo method, a recursive integration and as the winning probabilities are dependent to the points of the picked players, the process is sequential but the range of θ can be parallelized. Simulation collects the points of the players, normalizing the points with time, gives a histogram at a given range as output.

7. CONCLUSIONS

We studied multi-agent dynamics in two, three agent game models and we observed that the social regimes emerge from the ordering of the winning probabilities in these models. Two agent games yield two different societies. One is pure middle class society where the number of players with the highest score is the same as the players with the lowest scores. The second regime yields in an egalitarian society where all players share the same wealth. In three agent games, we studied six different societies which were introduced by Mungan and Rador [8]. On the back of methodologies introduced in two and three agent games, we studied a general $n+1$ agent game model and introduced an analytically solvable approach for this generalized model.

In all these models, a game is defined as a series of microscopic competitions where in each competition a finite number of players are randomly selected and ordered according to their points. A winning probability is assigned from a single probability set, to each player with respect to the ordering of the points. From this perspective we studied merger game model in three agent games. In this model there are two conflicting competition types. First is a competitive game and the second is merger game. By definition, merger games refer two agent games. Which led us to resolve three agent games in terms of two agent mini tournaments. The probability sets for each type of competitions are generated by the help of a single variable denoted by θ. From the core rules of the model we found that merger is effective at $\theta > 3/5$ and increasing θ means increasing the competitiveness of the game. $\theta = 1$ is considered as the extreme competitive limit. In three agent games the solution for this limit is represented in Equation (3.15).

The exact equation for the merger model is represented in Equation (5.5) and Equation (5.6). Unfortunately an analytical approach for these formulas is not apparent. Nevertheless with numerical analysis, we could understand some of the characteristics that came out from the model. By simulating the model in the range of θ, we observed the effect of merger games when the competitiveness of the game was increased.

In the effective range of θ, playing the game without a merger option mostly yields a C^- society which we explained in *section* 3.2.1. We observed that merger model yields C_S^+ society where a shock emerges.

For lower values of θ, game is totally dominated by merger games. Increasing the value of θ, we found that at some extreme values of θ the players at shock regions become isolated from merged players in lower regions and form a sub-society and other shocks emerge with a self similar structure. This behavior occur in two types of transitions. The θ values that transitions emerge are represented in Figure 5.6.

Analyzing the interactions between the sub-societies, we introduced the characteristic polynomials for two shock solution . As one can see in Figure 5.8,by borrowing one term from the simulation data we found that polynomials fit the shocks.

To observe the emergence of the shocks show a scaling behavior in the extreme competitive limit, we isolated the first shock and observed the system as two parts. We found out that scaling the system with Φ^n yields the shock locations scaled by Φ^{2n} and by taking the location of the first shock from the simulation data, we find Φ as "*golden ratio* $- 1$".

In extreme competitive limit, we observe that the game confirms the competitive pattern in Equation (3.15) but it forms sub-societies with a scaling behavior.

In conclusion, merger model leads to a significant increase in the competitiveness of the regimes. It would be interesting to apply this model to realistic data. This model can be applied to the cases where extreme competitiveness is observed.

APPENDIX A: SIMULATION SOURCE CODE

```
//  This program is a generic simulation  for n-player games
//  with merger option implemented only for three player
//  games.
//
//  Copyright (C) 2009  Rustu Derici <rustuderici@gmail.com>
//
//  This program is free software: you can redistribute it
//  and/or modify //  it under the terms of the GNU General
//  Public License as published by the Free Software
//  Foundation, either version 3 of the License, or  any
//  later version.
//
//  This program is distributed in the hope that it will
//  be useful, but WITHOUT ANY WARRANTY; without even the
//  implied warranty of MERCHANTABILITY or FITNESS FOR A
//  PARTICULAR PURPOSE.  See the GNU General Public
//  License for more details.
//
//  You may have  a copy of the GNU General Public License
//  version 3 from <http://www.gnu.org/licenses/gpl-3.0.txt>


#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_histogram.h>
#include <gsl/gsl_sort.h>
```

```
#include <gsl/gsl_block.h>

// function checks for the condition 1 ( y+z >= x )
// for a merger option. It returns true or false.
int merger_condition1( gsl_block_int *player_point,
                       gsl_block_int *agent_index,int n);

// function checks for the condition 2
// ( z >= (theoretical max points /2 )
// for a merger option. It returns true or false.
int merger_condition2( gsl_block_int *player_point,
                       gsl_block_int *agent_index,
                       int n,int N,int t);

// main condition checking function for merger
int is_merger( int condition, gsl_block_int *player_point,
               gsl_block_int *agent_index,int n,int N,
               int t);

// calculation of the modified probabilities
// for the merger option
int merger_prob_conf(double prob_merger_p2,
                     gsl_block *prob_merger);

// after game is finished the resulting histogram and some
// useful information is written to separate files with
// extension .out.dat and .info.dat respectively.
// Files are written to the same directory that program runs
int results( int HIST, gsl_block_int *player_point,
             gsl_block_int *player_playedgame,int N,
             int n,gsl_block *resul,int MergerCnt,
             int NoMergerCnt,int cond,double prob_merge,
```

```
             gsl_block *prob_default,gsl_block *prob_merger,
             int t);

// microscopic competition function
int play_game( gsl_block_int *player_point,
               gsl_block_int *agent_index,
               gsl_block *probs_shared,gsl_rng *r);



// probability deployment function
int prob_share( gsl_block *prob_default,int m,
                gsl_block *prob_shr,
                gsl_block_int *agent_index,
                gsl_block_int *player_point);

// descendingly sorting agents
int sort_agents(gsl_block_int *agent_index,
                gsl_block_int  *player_point,int n,int N);

//usage information of the agents
int usage(char * pname);

// randomly picking the agents
int select_agents(gsl_block_int *agent_index,
                 int n, int N, gsl_rng *r);



int main(int argc ,char **argv )
{
// loop variables
  int i,j;
// the counter for tour
```

```
  int tour_count=0;
// the counter for competitions in a tour
  int competition_count;
// normalization of the probabilities should give one
// this variable holds the sum of given probability arguments
  double prob_check;
// theta value for the probability modification for merger
  double prob_merge_p2;
// limit time for a game
  int TOUR;
// histogram range for the results
  int HIST;
//total player count
  int N;
// agent count that will initiate to a single competition
  int n;
// merger condition . 0 means no merger
  int condition ;
// minimum number of arguments before probs
  int noabp=7;
// count for merger games initiated in the whole played games
  int merger_cnt=0;
// count for merger games initiated in the whole played games
  int nomerger_cnt=0;
// time variable
  int t=0;
// determines if it is a merger game or not
  int mflag;

//SETUP GSL RANDOM VARIABLES

  const gsl_rng_type * T;
```

```
  gsl_rng * r;
  gsl_rng_env_setup();
// we set a default seed for the game to analyze
// the differences in the same sequence of randomness.
  T = gsl_rng_default;

// random number generator variable
  r = gsl_rng_alloc (T);

//INITIALIZATION OF VARIABLES

  if (argc< noabp)
    {
      // if argument count is less than necessary print
      //  usage and exit the program
      usage(argv[0]);
    }

N=atoi(argv[1]);
n=atoi(argv[2]);
TOUR=atoi(argv[3]);
HIST=atoi(argv[4]);
condition=atoi(argv[5]);

// if condition 0 is set,meaning no merger game,
// we will not need a merger modifier
if (condition > 0 )
      prob_merge_p2=atof(argv[6]);
else
      noabp=6;

if ((argc-noabp)!=n  && (argc-noabp)!=0 )
```

```
{

    // if the given probability arguments are less than the players
    // initiating a competition, which is define by n, then print
    // this error ,usage information and exit the program
    printf("ERROR: you should give exactly same number of winning
                    possibilities with agent number \"n\" !!\n");
    usage(argv[0]);

}



// These are variables with gsl data types

// holds the shared probabilities during each  competition in a
// no merger game.
gsl_block *probs_shr=gsl_block_calloc (n);

// holds the shared probabilities during each  competition in a
//  merger game. Used for merger check, if the modified probabilities
//  are higher than actual ones for the merging players
gsl_block *probs_shr_mrg=gsl_block_calloc (n);

// the index of players that participate a competition
gsl_block_int *agents_index=gsl_block_int_calloc(n);

// holds the points of each player
gsl_block_int  *player_points=gsl_block_int_calloc (N);

// holds the count of competitions that an individual player initiated
gsl_block_int *player_gameplayed=gsl_block_int_calloc (N);
```

```
// holds given probability values for a no merger game
gsl_block *probs_default=gsl_block_calloc (n);

// resulting histogram of the game
gsl_block *res=gsl_block_calloc (HIST);

// holds calculated probability values for a merger game
gsl_block *probs_merger=gsl_block_calloc (n);



//Probability configuration and normalization check

    prob_check=0.0;

    if (argc==noabp && (condition > 0) && n==3 )
    {
     // this construction is the resolution of three agent games
     // in terms of two agent games

         probs_default->data[0]=(double)(prob_merge_p2*prob_merge_p2)
         +(double)(prob_merge_p2*(1.0-prob_merge_p2)/3.0);
         probs_default->data[1]=
      (double)(4.0*prob_merge_p2*(1.0-prob_merge_p2)/3.0);
 probs_default->data[2]=(double)((1.0-prob_merge_p2)
             *(1.0-prob_merge_p2))
             *(+(double)(prob_merge_p2*(1.0-prob_merge_p2)/3.0);
 prob_check=probs_default->data[0]+probs_default->data[1]
             +probs_default->data[2];
    }
    else if((argc-noabp)==n  )
    {
        for (i=0;i<n;i++)
```

```
        {
          // setting the default winning probabilities of a no merger game
          // and gets their total sum for a check
          probs_default->data[i]=(double)atof(argv[i+noabp]);
          prob_check+=probs_default->data[i];
        }
    }

    if (((double)prob_check < (double)0.999)
         || ((double)prob_check > (double)1.0001))
    {
        //sum of  given probabilities are not equal to one..
        //give an error and usage info and exit.
        printf("ERROR: sum of winning possibilities
                          should be equal to 1 !!\n");
        printf("       your sum is : %lf  \n",prob_check);
        usage(argv[0]);
     }

// setting the modified  winning probabilities for
     if (condition > 0)
        merger_prob_conf(prob_merge_p2,probs_merger);

//GAME  STARTS

while (tour_count < TOUR )
{
    t++; // counting time
    competition_count=0; // reset competition counter
    while (competition_count < N/n)
    {
      // unit time is N/n competitions
```

```
//reset merger flag
// default action is always no merger.
// merging is conditional
mflag=0;
select_agents(agents_index,n,N,r);

sort_agents(agents_index,player_points,n,N);
// probabilities for a no merger game is set
prob_share(probs_default,n,probs_shr,
           agents_index,player_points);

 if (is_merger(condition,player_points,agents_index,n,N,t))
 {
       //if game is verified as a merger game ,
       //modified winning probabilities are calculated.
       prob_share(probs_merger,n,probs_shr_mrg,
                   agents_index,player_points);

       //as merger option is set only to three agent games
       //we can make controls over indexes without
       //generalization this control below checks that
       //if the modified probabilities are higher than
       //the actual probabilities for merging players
      if ( ((double)probs_shr_mrg->data[1]
          > (double)probs_shr->data[1])
                  &&
              ((double)probs_shr_mrg->data[2]
          > (double)probs_shr->data[2])
         )
        {
           mflag=1;
        }
```

```
            else
             {
                mflag=0;
             }

        }
        //play is separated from is_merger check for there is also "else"
        //option for is_merger where we need not to implement,
        // as mflag = 0 by default ,and we reset mflag
        // at the beginning of every competition.
        if (mflag==0) // this is a no merger game
        {
                nomerger_cnt++;
                play_game(player_points,agents_index,probs_shr,r);
        }
        else  // this is a merger game
        {
              merger_cnt++;
              play_game(player_points,agents_index,probs_shr_mrg,r);
         }



      for (j=0;j<n;j++)
      {
          // increase the initiated game count for the players that
          // participate the competition.
           player_gameplayed->data[agents_index->data[j]]++;
       }

  competition_count++;
  //competition ends here
  }
```

```
tour_count ++;
// tour ends here
}

//generating the histogram
results(HIST,player_points,player_gameplayed,
        N,n,res,merger_cnt,nomerger_cnt,condition,
        prob_merge_p2,probs_default,probs_merger,t);

gsl_rng_free (r);
gsl_block_free(probs_shr);
gsl_block_free(probs_shr_mrg);
gsl_block_int_free(agents_index);
gsl_block_int_free(player_points);
gsl_block_int_free(player_gameplayed);
gsl_block_free(probs_default);
gsl_block_free(probs_merger);
gsl_block_free(res);
return 0;
}

int merger_condition1( gsl_block_int *player_point,
                       gsl_block_int *agent_index,int n)
{
    if (n!=3)
          return 0; //function is called only in three agent games
    else
    {     // if the sum of middle and low points are equal
    {     // or higher than high points there will be a merger
          if(  player_point->data[agent_index->data[1]]
               +player_point->data[agent_index->data[2]]
```

```
                >=player_point->data[agent_index->data[0]]
               )
              {
                      return 1;
              }
            else
                      return 0;
    }

}

int merger_condition2( gsl_block_int *player_point,
                       gsl_block_int *agent_index,int n,int N,int t)
{

    if (n!=3)
            return 0; //function is called only in three agent games
    else
     {
         // if lowest point is greater than the
         // (theoretical maximum points)/2, theoretical max=time
         if ((double)player_point->data[agent_index->data[2]]
             >=(double)((double)(t*1.0)/(double)2.0))
          {
            return 1;
          }
  else
            return 0;
      }
}

 int is_merger(int condition, gsl_block_int *player_point,
```

```
              gsl_block_int *agent_index,int n,int N,int t)
{
     int res=0;
        // if merger game is selected at the arguments
        // of the program and it is a competitive game
        //we check  for the convenient conditioning for
        // merger game
       if ( condition != 0
                      &&
              player_point->data[agent_index->data[1]]
           <  player_point->data[agent_index->data[0]]
                     &&
              player_point->data[agent_index->data[2]]
           <= player_point->data[agent_index->data[1]]
          )
          {
               switch (condition)
               {
                 case 1:  //calling for condition 1 check
                    res=merger_condition1(player_point,agent_index,n);
                 break;
                 case 2:   //calling for condition2 check
                    res=merger_condition2(player_point,agent_index,n,N,t);
                 break;
                 default:
                    res=0; // default act is no merger.
                }
          }
        else
               res=0;

return res;
```

```
 }

int merger_prob_conf(double prob_merger_p2,gsl_block *prob_merger)
{
  double q2;
  double p2;
  p2=prob_merger_p2;
  q2=1.0-p2;

  prob_merger->data[0]=q2;
  prob_merger->data[1]=p2*p2;
  prob_merger->data[2]=p2*q2;

return 0;
}

int results(int HIST, gsl_block_int *player_point,
            gsl_block_int *player_playedgame,int N,
            int n,gsl_block *resul,int MergerCnt,int NoMergerCnt,
            int cond,double prob_merge,gsl_block *prob_default,
            gsl_block *prob_merger,int t)
{



  int i,j;
  int maxpoint;
  double percentageMerger=0.0;
  double percentageNoMerger=0.0;
  double totalgames=0.0;
  double gamecount=0.0;
  char filename[1024];
  char filename2[1024];
```

```
char buffer[100];
FILE *fp;  // file for output histogram
FILE *fp2; // file for statistical information about the game

// generating filenames
sprintf(filename,"%dplayer_%dagent",N,n);
sprintf(filename2,"%dplayer_%dagent",N,n);
for (i=0;i<n;i++)
{  // default probabilities set to the game
  sprintf(buffer,"_%lf",prob_default->data[i]);
  strcat(filename,buffer);
  strcat(filename2,buffer);
}

if (cond==0)
{
   strcat(filename,"_nomerger");
   strcat(filename2,"_nomerger");
}
else
{
   sprintf(buffer,"_merger_condition%d_theta%lf",cond,prob_merge);
   strcat(filename,buffer);
   strcat(filename2,buffer);
   // if it is a merger game give the merger option
   // modified probabilities
   for (i=0;i<n;i++)
    {
       sprintf(buffer,"_%lf",prob_merger->data[i]);
       strcat(filename,buffer);
       strcat(filename2,buffer);
     }
```

```
  }
    //giving file extensions
   strcat(filename,"_out.dat");
   strcat(filename2,"_info.dat");



  // open files
  fp=fopen(filename,"w");
  fp2=fopen(filename2,"w");
  //allocate histogram
  gsl_block *resu=gsl_block_calloc (HIST);
  // the maximum points that a player can gain is
  // theoretically equal to time
  maxpoint=t;

 gsl_histogram *h = gsl_histogram_alloc(HIST);
 gsl_histogram_set_ranges_uniform(h, 0, maxpoint);

//preparing the histogram
  for(i=0;i<N;i++)
        gsl_histogram_increment(h,(double)player_point->data[i]);

//getting the histogram
  for(i=0;i<HIST;i++)
    for(j=0;j<i;j++)
        resu->data[i]+=gsl_histogram_get(h,j);

  gsl_histogram_free (h);
  percentageMerger=(double)MergerCnt/(double)(MergerCnt+NoMergerCnt);
  percentageNoMerger=(double)NoMergerCnt/(double)(MergerCnt+NoMergerCnt);
  for(i=0;i<N;i++) gamecount+=(double)player_point->data[i]*1.0;
```

```
// writing the statistical data of the game to info.dat file
fprintf(fp2,"GameProbs:%2.3lf-%2.3lf-%2.3lf:",
        prob_default->data[0],
        prob_default->data[1],
        prob_default->data[2]);

fprintf(fp2,"MergeProb(P''):%2.3lf:",prob_merge);

fprintf(fp2,"MergingProbs:%2.3lf-%2.3lf-%2.3lf:",
        prob_merger->data[0],
        prob_merger->data[1],
        prob_merger->data[2]);

fprintf(fp2,"Total Players---:%d:",N);
fprintf(fp2,"Total Agents----:%d:",n);
fprintf(fp2,"Game MergeCond--:%d:",cond);
fprintf(fp2,"Total GamePlayed:%d:",(int)gamecount);
fprintf(fp2,"Theoretical Max-:%d:",maxpoint);

fprintf(fp2,"Merger Played---:%d:%2.3lf:",
         MergerCnt,percentageMerger*100.0);

fprintf(fp2,"NoMerger Played-:%d:%2.3lf:",
        NoMergerCnt,percentageNoMerger*100.0);

fprintf(fp2,"Total Agents----:%d:",n);
fprintf(fp2,"Total Players---:%d",N);



//writing the histogram to out.dat file
for(i=0;i<HIST;i++)
```

```
    fprintf(fp,"%lf %lf \n",(double) (i*1.0)/HIST,resu->data[i]/N);


fclose(fp2);
fclose(fp);
gsl_block_free(resu);
return 0;
}



int play_game( gsl_block_int *player_point,
               gsl_block_int *agent_index,
               gsl_block *probs_shared, gsl_rng *r)
 {
   double game;
   int i;
   double gamechk;

   //a random number is generated
   game=gsl_rng_uniform(r);
   i=0;

   gamechk=probs_shared->data[0];

   while(game> gamechk)
   {
    // to find the range that the random number is,
    //  we line up the probs of descendingly sorted agents
     i=i+1;
     gamechk+=probs_shared->data[i];

    }
```

```
    player_point->data[agent_index->data[i]]++;

return 0;
}



int prob_share( gsl_block *prob_default, int m,
                gsl_block *prob_shr,
                gsl_block_int *agent_index,
                gsl_block_int *player_point)
{
  double pcurrent;
  double ptmp;
  int pos=0;
  int count=0;
  int i;
  while (pos < m)
  {
      pcurrent=prob_default->data[pos];
      count=1;
  //as players are sorted we check the count of
  //equal points and share their probabilities equally
  //otherwise they get the preset value of probability.
     while(player_point->data[agent_index->data[pos]]
         ==player_point->data[agent_index->data[pos+1]])
     {
         if (pos < (m-1))
        {
          pos++;
          pcurrent+=prob_default->data[pos];
          count++;
```

```
        }
        else
           break;

     }

    for (i=(pos-count+1);i<=pos;i++)
    {
        ptmp=pcurrent/count;
        prob_shr->data[i]=ptmp;
    }
  pos++;
  }
return 0 ;
}



int sort_agents(gsl_block_int *agent_index,
                gsl_block_int  *player_point,int n,int N)
{
 int i;
 gsl_block_uint *agents_sorted_index=gsl_block_uint_calloc(n);
 gsl_block *agents_point=gsl_block_calloc (n);
 gsl_block_int *tmp_index=gsl_block_int_calloc(n);

 // as gsl sort engine requires double variables and
 //  we store them as integers.a conversion buffer is required
 for (i=0;i<n;i++)
  {
    agents_point->data[i]=
                 (double)player_point->data[agent_index->data[i]];
```

```
  }

//descendingly sorting agents
 gsl_sort_largest_index(agents_sorted_index->data,
                        n,agents_point->data,1,n);

 //rearranging the indexes
 for (i=0;i<n;i++)
 {
    tmp_index->data[i]=agent_index->data[agents_sorted_index->data[i]];
 }

 for (i=0;i<n;i++)
 {
    agent_index->data[i]=tmp_index->data[i];
 }
 gsl_block_int_free(tmp_index);
 gsl_block_uint_free(agents_sorted_index);
 gsl_block_free(agents_point);
return 0;
}

int usage(char * pname)
{
printf("USAGE : \n");
printf("if you will use MERGER OPTION (only in three agent games)\n\n");
printf("\t   %s  N n TOUR HIST merger_condition
              prob_merge_p2 p[1] p[2] ... p[n] \n\n",pname);

printf("or  \n\n");
printf("\t   %s  N n TOUR HIST merger_condition
                 prob_merge_p2  \n\n",pname);
```

```
printf("\t in this option probs will
               be calculated by means of two agent tournaments  \n\n");
printf("if you will play NO MERGER GAME
           you WILL NOT set prob_merge_p2 as: \n\n");
printf("\t   %s  N n TOUR HIST 0  p[1] p[2] ... p[n] \n\n",pname);
printf(" N: number of total players \n");
printf(" n: number of agents participating a single competition \n");
printf(" TOUR: game cycles ..for each turn agents play n games.\n");
printf(" HIST: Histogram range\n");
printf(" merger_condition:\n  0:no_merger \n
                                1: y+z>= x \n  \t
                                2: z>= MaxPoints(t)/2 \n");

printf(" prob_merge_p2: it is the p2 value for two
         agent simulated three agent game rules on merging \n");
printf(" p[1..n]: winning possibilities of selected agents.\n");
exit(1);

}




int select_agents(gsl_block_int *agent_index, int n,int N,gsl_rng *r)
{

 int i,j;
 int flag=0;
  // randomly picking the agents .
  // there should not be agents with the same index number .
  while (flag==0)
  {
    for ( i=0;i<n;i++)
```

```
    {
        agent_index->data[i]=gsl_rng_uniform_int(r,N);
    }

    flag=0;

    for (i=0;i<n-1;i++)
       for(j=i+1;j<n;j++)
       {
          if (agent_index->data[i]==agent_index->data[j])
          {
             flag=1;
             break;
           }
        }
       if (flag==1)
               flag=0;
       else
               flag=1;
  }
return 0;
}
```

REFERENCES

1. Matteo Marsili and Yi-Cheng Zhang, *"Interacting Individuals Leading to Zipf's Law"*, Phys. Rev. Lett. 80, 2741 - 2744, 1998.

2. E. Ben-Naim, P.L. Krapivsky, S. Redner, *"Bifurcations and Patterns in Compromise Processes"*, Physica D 183, 190, 2003.

3. E. Ben-Naim, F. Vazquez and S. Redner, *"Parity and Predictability of Competitions"*, Journal of Quantitative Analysis in Sports: Vol. 2 : Iss. 4, Article 1., 2006

4. E. Ben-Naim and S Redner, *"Dynamics of social diversity"*, J. Stat. Mech. L11002, 2005.

5. K.Malarz, D.Stauffer, K.Kulakowski, *"Bonabeau model on a fully connected graph"*, arXiv:physics/0502118v3 [physics.soc-ph] 2005

6. E. Ben-Naim, B. Kahng, and J.S. Kim, *"Dynamics of multi-player games"*, J. Stat. Mech. P07001, 2006

7. E. Ben-Naim, F. Vazquez, and S. Redner, *"On The Structure of Competitive Societies"*, Eur. Phys. Jour. B 49, 531, 2006

8. Tonguç Rador and Muhittin Mungan, *"Dynamics of Three Agent Games"*, J. Phys. A: Math. Theor. 41 055002, 2008

9. Makoto Matsumoto and Takuji Nishimura, *"Mersenne Twister: A 623- dimensionally equidistributed uniform pseudorandom number generator"*, ACM Transactions on Modeling and Computer Simulation, Vol. 8, No. 1, Pages 3−30, 1998.