A numerical study of fixation probabilities for strategies in the Iterated Prisoner's Dilemma

Marc Harper Vincent Knight

Abstract

The Iterated Prisoner's Dilemma is a well established framework for the study of emergent behaviour. In this paper an extensive numerical study of the evolutionary dynamics of this framework are presented.

Fixation probabilities for Moran processes are obtained for 172 different strategies. This is done in both a standard 200 turn interaction and a noisy setting.

To the authors knowledge this is the largest such study. It allows for insights about the behaviour and performance of strategies with regard to their survival in an evolutionary setting.

1 Introduction

Since the formulation of the Moran Process in [9], this model of evolutionary population dynamics has been used to gain insights about the evolutionary stability of strategies in a number of settings. Similarly since the first Iterated Prisoner's Dilemma (IPD) tournament described in [2] the Prisoner's dilemma has been used to understand the evolution of cooperative behaviour in complex systems.

The analytical models of a Moran process are based on the relative fitness between two strategies and take this to be a fixed value r [11]. This is a valid model for simple strategies of the Prisoner's Dilemma such as to *always cooperate* or *always defect*. This manuscript provides a detailed numerical analysis of **172** complex and adaptive strategies for the IPD. In this case the relative fitness of a strategy is dependent on the population distribution.

Further deviations from the analytical model occur when interactions between players are subject to uncertainty. This is referred to as noise and has been considered in the IPD setting in [4, 10, 14]. Noise is also considered here.

This work provides answers to the following questions:

1. What strategies are good invaders?
2. What strategies are good at resisting invasion?
3. How does the population size affect these findings?

Figure 1 shows a diagrammatic representation of the Moran process. The Moran process is a stochastic birth death process on a finite population in which the population size stays constant over time. Individuals are **selected** according to a given fitness landscape. Once selected, a given individual is reproduced and similarly another individual is chosen to be removed from the population. In some settings mutation is also considered but without mutation (the case considered in this work) this process will arrive at an absorbing state where the population is entirely made up of a single individual. The probability with which a given strategy is the survivor is called the absorption probability. A more detailed analytic description of this is given in Section 3.

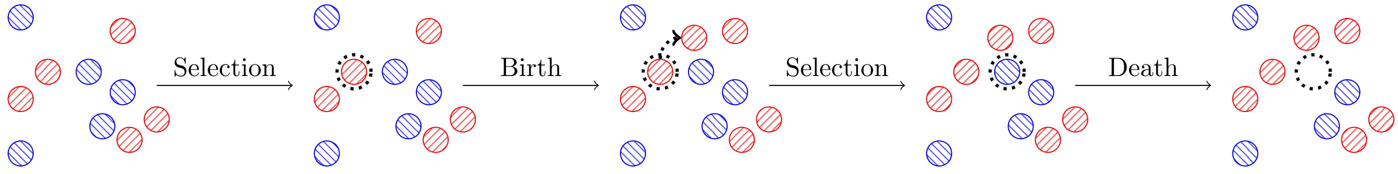


Figure 1: A diagrammatic representation of a Moran process

The Moran process was initially introduced in [9] in a genetic setting. It has sine been used in a variety of settings including the understanding of the spread of cooperative behaviour. However, as stated before, these mainly consider non sophisticated strategies. Some work has looked at evolutionary stability of strategies within the Prisoner's Dilemma [7] but this is not done in the more widely used setting of the Moran process but in terms of infinite population stability. In [3] Moran processes are looked at in a theoretic framework for a small subset of strategies. In [6] machine learning techniques are used to train a strategy capable of resisting invasion and also invade any memory one strategy.

The contribution of this work is a detailed and extensive analysis of absorption probabilities for 172 strategies. These strategies and the numerical simulations are from [12] which is an open source research library written for the study of the IPD. The strategies and simulation frameworks are automatically tested in accordance to best practice. The large number of strategies are available thanks to the open source nature of the project with over 40 contributions made by different programmers.

Section 2 will explain the methodological approach used, Section 3 will validate the methodology by comparing simulated results to analytical results. The main results of this manuscript are presented in Section 4 which will present a detailed analysis of all the data generated. Finally, Section 5 will conclude and offer future avenues for the work presented here.

2 Methodology

To carry out this large numerical experiment 172 strategies are used from [12]. These include 169 default strategies in the library at the time (excluding strategies classified as having a long run time) as well as the following 3 finite state machine machine strategies [1]:

Appendix A shows all the players in question. More information about each player can be obtained in the documentation for [12]. The memory depth of the used strategies is shown in Table 1a.

Memory Depth	0	1	2	3	4	5	6	9	10	11	12	16	20	40	200	∞
Count	3	31	12	8	2	6	1	1	5	1	1	2	2	2	1	94

(a) Memory depth

Stochastic	Count
False	123
True	49

(b) Stochastic versus deterministic

Table 1: Summary of properties of used strategies

All strategies are paired and these pairs are used in 1000 repetitions of a Moran process assuming a starting population of $(N/2, N/2)$. This is repeated for even N between 2 and 14. The fixation probability is then estimated for each value of N.

Note that due to the high computational cost of these experiments, for any given interaction between two players within the Moran process the outcome is sampled from a pre computed cache of 1000 match outcomes. This is carried out using the approximate Moran process implemented in [12].

As an example, Figure 2 shows the scores between two players that over the 1000 outcomes gives 971 different scores.

A variety of software libraries have been used in this work:

- The Axelrod library (IPD strategies and Moran processes) [12].
- The matplotlib library (visualisation) [5].
- The pandas and numpy libraries (data manipulation) [8, 13].

Section 3 will validate this approach against theoretic results.

3 Validation

As described in [11] Consider the payoff matrix:

$$M = \begin{pmatrix} a, b \\ c, d \end{pmatrix} \tag{1}$$

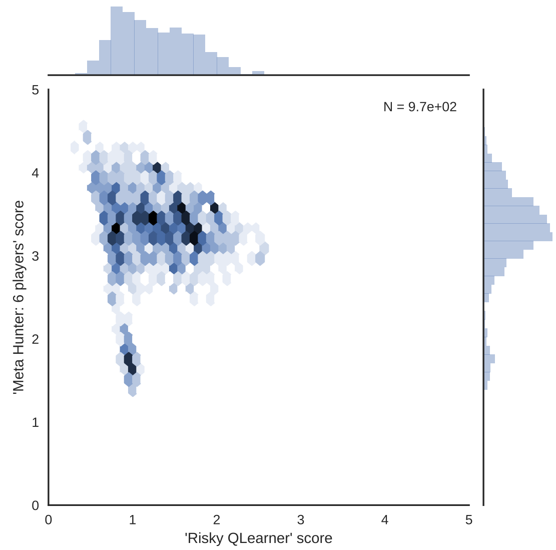


Figure 2: All possible scores for the pair of strategies that have the most different number of match outcomes

The expected payoffs of i players of the first type in a population with $N-i$ players of the second type are given by:

$$F_i = \frac{a(i-1)+b(N-i)}{N-1} \tag{2}$$

$$G_i = \frac{ci+d(N-i-1)}{N-1} \tag{3}$$

With an intensity of selection ω the fitness of both strategies is given by:

$$f_i = 1-\omega+\omega F_i \tag{4}$$

$$g_i = 1-\omega+\omega G_i \tag{5}$$

The transitions within the birth death process that underpins the Moran process are then given by:

$$p_{i,i+1} = \frac{if_i}{if_i+(N-i)g_i}\frac{N-i}{N} \tag{6}$$

$$p_{i,i-1} = \frac{(N-i)g_i}{if_i+(N-i)g_i}\frac{i}{N} \tag{7}$$

$$p_{ii} = 1-p_{i,i+1}-p_{i,i-1} \tag{8}$$

Using this it is a known result that the fixation probability of the first strategy in a population of i individuals of the first type (and $N-i$ individuals of the second. We have:

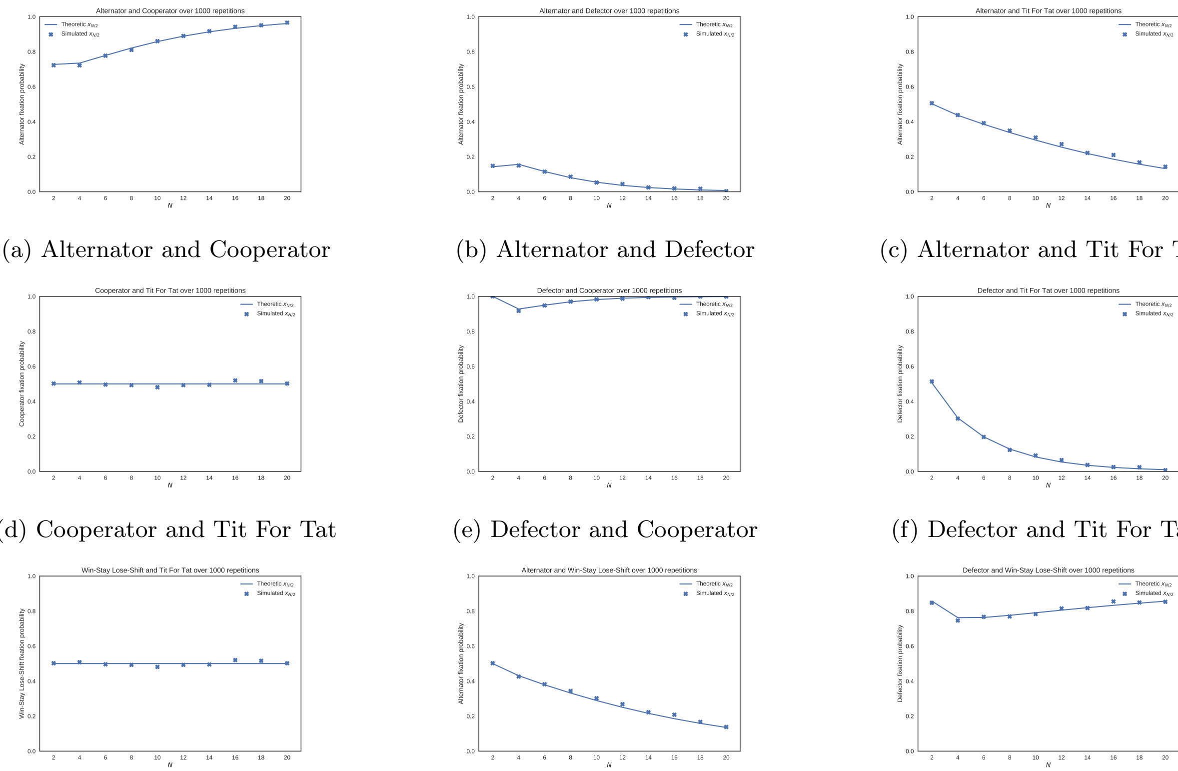
$$x_i = \frac{1+\sum_{j=1}^{i-1}\prod_{k=1}^{j}\gamma_j}{1+\sum_{j=1}^{N-1}\prod_{k=1}^{j}\gamma_j} \tag{9}$$

where:

$$\gamma_j = \frac{p_{j,j-1}}{p_{j,j+1}}$$

Using this comparisons of $x_{N/2}$ are shown in Figure 3. Note that these are all deterministic strategies and show a perfect match up between the expected value of (9) and the actual Moran process for all strategies pairs.

Figure 4 shows the fixation probabilities for stochastic strategies. These are no longer a good match which highlights the weakness of the analytical formulae that relies on the average payoffs. A detailed analysis of the 172 strategies considered will be shown in the next Section.



(a) Alternator and Cooperator (b) Alternator and Defector (c) Alternator and Tit For Tat

(d) Cooperator and Tit For Tat (e) Defector and Cooperator (f) Defector and Tit For Tat

(g) Win Stay Lose Shift and Tit For Tat (h) Alternator and Win Stay Lose Shift (i) Defector and Win Stay Lose Shift

Figure 3: Comparison of theoretic and actual Moran Process fixation probabilities for **deterministic** strategies

4 Numerical results

5 Conclusion

Further work:

- Spatial structure;
- More than two types in the population;
- Modified Moran processes (Fermi selection);
- Mutation;

Acknowledgements

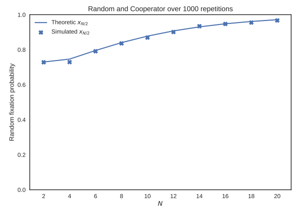
This work was performed using the computational facilities of the Advanced Research Computing @ Cardiff (ARCCA) Division, Cardiff University.

References

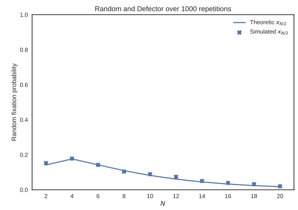
[1] Wendy Ashlock and Daniel Ashlock. "Changes in Prisoner ' s Dilemma Strategies Over Evolutionary Time With Different Population Sizes". In: (2006), pp. 1001–1008.

[2] R. Axelrod. "Effective Choice in the Prisoner's Dilemma". In: *Journal of Conflict Resolution* 24.1 (1980), pp. 3–25.

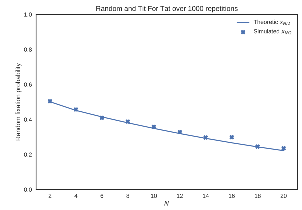
[3] Seung Ki Baek et al. "Comparing reactive and memory- one strategies of direct reciprocity". In: *Nature Publishing Group* (2016), pp. 1–13. DOI: 10.1038/srep25676. URL: http://dx.doi.org/10.1038/srep25676.



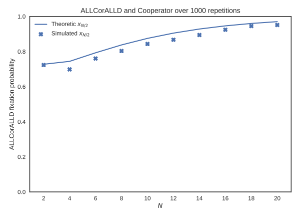
(a) Random and Cooperator



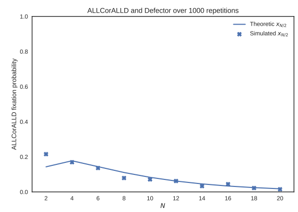
(b) Random and Defector



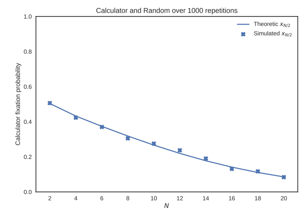
(c) Random and Tit For Tat



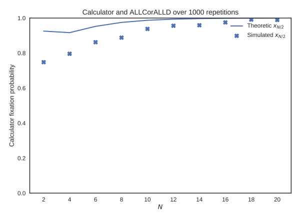
(d) All C or all D and Cooperator



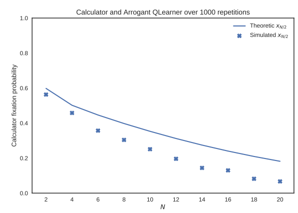
(e) All C or all D and Defector



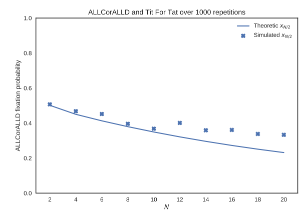
(f) Calculator and Random



(g) Calculator and All C or all D



(h) Calculator and Arrogant Q learner



(i) All C or all D and Tit For Tat

Figure 4: Comparison of theoretic and actual Moran Process fixation probabilities for **stochastic** strategies

[4] Jonathan Bendor. "Uncertainty and the Evolution of Cooperation". In: *The Journal of Conflict Resolution* 37.4 (1993), pp. 709–734.

[5] John D Hunter. "Matplotlib: A 2D graphics environment". In: *Computing In Science & Engineering* 9.3 (2007), pp. 90–95.

[6] Christopher Lee, Marc Harper, and Dashiell Fryer. "The Art of War: Beyond Memory-one Strategies in Population Games". In: *Plos One* 10.3 (2015), e0120625. ISSN: 1932-6203. DOI: `10.1371/journal.pone.0120625`. URL: `http://dx.plos.org/10.1371/journal.pone.0120625`.

[7] Jiawei Li, Graham Kendall, and Senior Member. "The effect of memory size on the evolutionary stability of strategies in iterated prisoner ' s dilemma". In: X.X (2014), pp. 1–8.

[8] Wes McKinney et al. "Data structures for statistical computing in python". In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. van der Voort S, Millman J. 2010, pp. 51–56.

[9] P.A.P. Moran. "Random Processes in Genetics". In: April (1957), pp. 60–71.

[10] M Nowak and K Sigmund. "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game." In: *Nature* 364.6432 (1993), pp. 56–58. ISSN: 0028-0836. DOI: `10.1038/364056a0`.

[11] Martin A Nowak. *Evolutionary Dynamics: Exploring the Equations of Life*. Cambridge: Harvard University Press. ISBN: 0674023382. DOI: `10.1086/523139`.

[12] The Axelrod project developers. *Axelrod: v2.9.0*. Apr. 2016. DOI: `499122`. URL: `http://dx.doi.org/10.5281/zenodo.499122`.

[13] Stfan van der Walt, S Chris Colbert, and Gael Varoquaux. "The NumPy array: a structure for efficient numerical computation". In: *Computing in Science & Engineering* 13.2 (2011), pp. 22–30.

[14] Jianzhong Wu and Robert Axelrod. "How to Cope with Noise in the Iterated Prisoner's Dilemma". In: *The Journal of Conflict Resolution* 39.1 (1995).

A List of players

1. Adaptive
2. Adaptive Tit For Tat: 0.5
3. Aggravater
4. ALLCorALLD
5. Alternator
6. Alternator Hunter
7. AntiCycler
8. Anti Tit For Tat
9. Adaptive Pavlov 2006
10. Adaptive Pavlov 2011
11. Appeaser
12. Arrogant QLearner
13. Average Copier
14. Better and Better
15. BackStabber: ('D', 'D')
16. Bully
17. Calculator
18. Cautious QLearner
19. Champion
20. CollectiveStrategy
21. Contrite Tit For Tat
22. Cooperator
23. Cooperator Hunter
24. Cycle Hunter
25. Cycler CCCCCD
26. Cycler CCCD
27. Cycler CCD
28. Cycler DC
29. Cycler DDC
30. Cycler CCCDCD
31. Davis: 10
32. Defector
33. Defector Hunter
34. Desperate
35. DoubleCrosser: ('D', 'D')
36. Doubler
37. EasyGo
38. Eatherley
39. Eventual Cycle Hunter
40. Evolved ANN
41. Evolved ANN 5
42. Evolved ANN 5 Noise 05
43. Evolved FSM 4
44. Evolved FSM 16
45. Evolved FSM 16 Noise 05
46. EvolvedLookerUp1_1_1
47. EvolvedLookerUp2_2_2
48. Evolved HMM 5
49. Feld: 1.0, 0.5, 200
50. Firm But Fair
51. Fool Me Forever
52. Fool Me Once
53. Forgetful Fool Me Once: 0.05
54. Forgetful Grudger
55. Forgiver
56. Forgiving Tit For Tat
57. Fortress3
58. Fortress4
59. GTFT: 0.33
60. General Soft Grudger: n=1,d=4,c=2
61. Soft Go By Majority
62. Soft Go By Majority: 10
63. Soft Go By Majority: 20
64. Soft Go By Majority: 40
65. Soft Go By Majority: 5
66. ϕ
67. Gradual
68. Gradual Killer: ('D', 'D', 'D', 'D', 'D', 'C', 'C')
69. Grofman
70. Grudger
71. GrudgerAlternator
72. Grumpy: Nice, 10, -10
73. Handshake
74. Hard Go By Majority
75. Hard Go By Majority: 10
76. Hard Go By Majority: 20
77. Hard Go By Majority: 40
78. Hard Go By Majority: 5
79. Hard Prober
80. Hard Tit For 2 Tats
81. Hard Tit For Tat
82. Hesitant QLearner
83. Hopeless
84. Inverse
85. Inverse Punisher
86. Joss: 0.9
87. Knowledgeable Worse and Worse
88. Level Punisher
89. Limited Retaliate: 0.1, 20
90. Limited Retaliate 2: 0.08, 15
91. Limited Retaliate 3: 0.05, 20
92. Math Constant Hunter
93. Naive Prober: 0.1
94. MEM2
95. Negation
96. Nice Average Copier
97. Nydegger
98. Omega TFT: 3, 8
99. Once Bitten

100. Opposite Grudger
101. π
102. Predator
103. Prober
104. Prober 2
105. Prober 3
106. Prober 4
107. Pun1
108. PSO Gambler 1_1_1
109. PSO Gambler 2_2_2
110. PSO Gambler 2_2_2 Noise 05
111. PSO Gambler Mem1
112. Punisher
113. Raider
114. Random: 0.5
115. Random Hunter
116. Remorseful Prober: 0.1
117. Resurrection
118. Retaliate: 0.1
119. Retaliate 2: 0.08
120. Retaliate 3: 0.05
121. Revised Downing: True
122. Ripoff
123. Risky QLearner
124. SelfSteem
125. ShortMem
126. Shubik
127. Slow Tit For Two Tats
128. Slow Tit For Two Tats 2
129. Sneaky Tit For Tat
130. Soft Grudger
131. Soft Joss: 0.9
132. SolutionB1
133. SolutionB5
134. Spiteful Tit For Tat
135. Stalker: D
136. Stochastic Cooperator
137. Stochastic WSLS: 0.05
138. Suspicious Tit For Tat
139. Tester
140. ThueMorse
141. ThueMorseInverse
142. Thumper
143. Tit For Tat
144. Tit For 2 Tats
145. Tricky Cooperator
146. Tricky Defector
147. Tullock: 11
148. Two Tits For Tat
149. VeryBad
150. Willing
151. Winner12
152. Winner21
153. Win-Shift Lose-Stay: D
154. Win-Stay Lose-Shift: C
155. Worse and Worse
156. Worse and Worse 2
157. Worse and Worse 3
158. ZD-Extort-2: 0.1111111111111111, 0.5
159. ZD-Extort-2 v2: 0.125, 0.5, 1
160. ZD-Extort-4: 0.23529411764705882, 0.25, 1
161. ZD-GTFT-2: 0.25, 0.5
162. ZD-GEN-2: 0.125, 0.5, 3
163. ZD-SET-2: 0.25, 0.0, 2
164. e
165. Meta Hunter: 6 players
166. Meta Hunter Aggressive: 7 players
167. Meta Majority Memory One: 31 players
168. Meta Winner Memory One: 31 players
169. NMWE Memory One: 31 players
170. FSM Player: [(0, ’C’, 7, ’C’), (0, ’D’, 1, ’C’), (1, ’C’, 11, ’D’), (1, ’D’, 11, ’D’), (2, ’C’, 8, ’D’), (2, ’D’, 8, ’C’), (3, ’C’, 3, ’C’), (3, ’D’, 12, ’D’), (4, ’C’, 6, ’C’), (4, ’D’, 3, ’C’), (5, ’C’, 11, ’C’), (5, ’D’, 8, ’D’), (6, ’C’, 13, ’D’), (6, ’D’, 14, ’C’), (7, ’C’, 4, ’D’), (7, ’D’, 2, ’D’), (8, ’C’, 14, ’D’), (8, ’D’, 8, ’D’), (9, ’C’, 0, ’C’), (9, ’D’, 10, ’D’), (10, ’C’, 8, ’C’), (10, ’D’, 15, ’C’), (11, ’C’, 6, ’D’), (11, ’D’, 5, ’D’), (12, ’C’, 6, ’D’), (12, ’D’, 9, ’D’), (13, ’C’, 9, ’D’), (13, ’D’, 8, ’D’), (14, ’C’, 8, ’D’), (14, ’D’, 13, ’D’), (15, ’C’, 4, ’C’), (15, ’D’, 5, ’C’)], 1, C
171. FSM Player: [(0, ’C’, 13, ’D’), (0, ’D’, 12, ’D’), (1, ’C’, 3, ’D’), (1, ’D’, 4, ’D’), (2, ’C’, 14, ’D’), (2, ’D’, 9, ’D’), (3, ’C’, 0, ’C’), (3, ’D’, 1, ’D’), (4, ’C’, 1, ’D’), (4, ’D’, 2, ’D’), (5, ’C’, 12, ’C’), (5, ’D’, 6, ’C’), (6, ’C’, 1, ’C’), (6, ’D’, 14, ’D’), (7, ’C’, 12, ’D’), (7, ’D’, 2, ’D’), (8, ’C’, 7, ’D’), (8, ’D’, 9, ’D’), (9, ’C’, 8, ’D’), (9, ’D’, 0, ’D’), (10, ’C’, 2, ’C’), (10, ’D’, 15, ’C’), (11, ’C’, 7, ’D’), (11, ’D’, 13, ’D’), (12, ’C’, 3, ’C’), (12, ’D’, 8, ’D’), (13, ’C’, 7, ’C’), (13, ’D’, 10, ’D’), (14, ’C’, 10, ’D’), (14, ’D’, 7, ’D’), (15, ’C’, 15, ’C’), (15, ’D’, 11, ’D’)], 1, C
172. FSM Player: [(0, ’C’, 0, ’C’), (0, ’D’, 3, ’C’), (1, ’C’, 5, ’D’), (1, ’D’, 0, ’C’), (2, ’C’, 3, ’C’), (2, ’D’, 2, ’D’), (3, ’C’, 4, ’D’), (3, ’D’, 6, ’D’), (4, ’C’, 3, ’C’), (4, ’D’, 1, ’D’), (5, ’C’, 6, ’C’), (5, ’D’, 3, ’D’), (6, ’C’, 6, ’D’), (6, ’D’, 6, ’D’), (7, ’C’, 7, ’D’), (7, ’D’, 5, ’C’)], 1, C