Delft University of Technology, Bachelor Seminar of Computer Science and Engineering

Unbeatable cooperative strategies under noise

Raymond Timmermans

June 27, 2021

Abstract

This paper combines the concepts of noise and stability in the iterated prisoner's dilemma. The purpose is to find strategies that score well (have a high cooperation percentage) in a homogeneous noisy environment, but are also robust to invasions. Generally, strategies that perform best under noise are maximally forgiving, however, susceptibility to exploitation then also increases. By modifying the payoff variables to increase the incentive for cooperation stability is more easily achieved. Performance and stability for several strategies under noise are compared.

1 Introduction

The evolution of cooperative behavior is widely studied in numerous areas including sociology, biology, and computer science. Recognizing under what conditions cooperative behavior can arise, thrive, and be stable is useful in many parts of society.

This paper uses the prisoner's dilemma as a model to draw conclusions. A single iteration of the prisoner's dilemma is defined as a game between two players. Both have the option to either "cooperate" or "defect". The payoff variables are as follows:

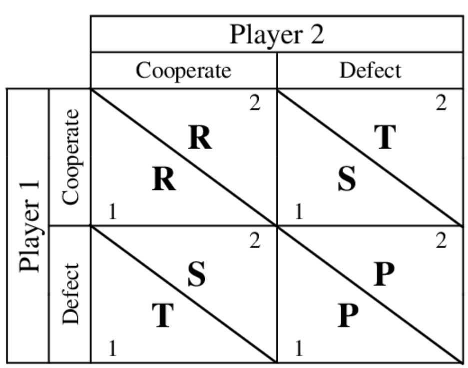

Figure 1: Payoff matrix for the Prisoner's Dilemma game.[12]

With additional constraints that $T > R > P > S$ and $2R > T + S$. The last constraint makes sure that it is more rewarding for both players to always cooperate than to alternate between cooperating and defecting. For a single iteration it is optimal to always defect, but in the iterated variant (IPD) cooperation might emerge [2, 4], which maximizes the average payoff.

Two additional concepts are used in this paper; noise and evolutionary stability. Noise is defined as a nonzero probability that the chosen "cooperate" will be changed into "defect" and vice versa. Strategies that normally do well in the IPD may perform poorly under noise [9, 10, 3, 5, 8]. For example, the well-known strategy tit for tat (TFT), where a player starts with cooperating and then copies the opponent's last move, performs excellent against itself (cooperation rate of 100%), but performance drops drastically when even just a slight bit of noise is introduced to the environment (cooperation rate of 50%). Jianzhong Wu and Robert Axelrod suggest that contrite or generous strategies can cope with noise better[13]. Robert Boyd explains the workings of contrite tit for tat (CTFT) nicely: "An individual is always in good standing on the first turn. It remains in good standing as long as it co-operates when CTFT specifies that it should co-operate. If an individual is not in good standing it can get back in good standing by co-operating on one turn. Then CTFT specifies that an individual should co-operate (i) if it is not in good standing, or (ii) if its opponent is in good standing; otherwise the individual should defect.[7]" Generous strategies include generous tit for tat (GTFT). GTFT is similar to TFT, but will not retaliate to a single defection some percentage of the time to prevent indefinite retaliation. Under noise, the expected value of two generous players playing against each other increases as their generosity increases, but so does their overall exploitability to others. A friendly environment where everyone cooperates is desirable, but it also needs to be robust to invasion; protection is needed against exploiters for the friendly environment to last.

For a strategy to be robust to invasions, it must be 'stable'. But there has been a lot of unclarity surrounding the word 'stability'. Jonathan Bendor and Piotr Swistak [6] clarified the confusion and explained the subtle differences between collective stability, weak evolutionary stability, and strong evolutionary stability. However, since noise and cooperation incentive, which I will get to in the next section, are continuous variables the differences between the three terms is not important for this paper. Therefore, stability in this paper refers to collective stability, which is the simplest of the three terms. Axelrod [2] defined collective stability as follows: If $V(i, j)$ is defined as the expected payoff for strategy i when playing against strategy j, then strategy i is collectively stable if for all strategies j, $V(i, i) >= V(j, i)$.

This paper combines the notions of noise and stability. Both topics have been widely studied, but not in combination. The objective is to find strategies that have a high cooperation percentage in a homogeneous noisy environment, but would also be stable.

A precise definition of stability will follow later, but it was found that CTFT is always stable and performs reasonably well. However, if the payoff variables are such that cooperation is incentivized enough, strategies such as GTFT can be stable as well and perform better.

The outline of this paper is as follows. Section 2 describes the general approach that was taken to analyze a particular strategy followed by the application of this approach to each to be analyzed strategy in more depth in section 3. In section 4 results are shown, which will be discussed in section 5 in more detail. Main conclusions and future research is discussed in section 6.

2 Methodology

Stability can be broken down into two separate categories. Stability against exploiters, and stability against even more generous strategies. For example, TFT is not exploitable, even under noise (proof will follow in section 3.1). However, the cooperator, a much more generous strategy than TFT, can still invade the population. At first glance, this does not seem like a problem, because more cooperation is achieved. However, at this point, the environment has changed into a heterogeneous and more generous environment and perhaps it would be susceptible to exploitation now. In this specific case, when introducing the defector to this environment, TFT will still triumph as it performs well against both strategies, whereas the cooperator and the defector do not. But this might not be always the case depending on the exact population, invader, and environment configuration. Evaluating stability in its fullest sense is complex, but a first step, which this paper solely focuses on is stability against exploiters. More precisely, a strategy 'i' is not exploitable if for all strategies 'j', such that $V(j,i) > V(i,j)$, then $V(i,i) >= V(j,i)$. In other words, all strategies that try to exploit i, will not be able to exploit it enough to invade the population. The only way then to invade the population is by cooperating more than the native population, which is not the focus of this paper. From this point on, stability is referred to as stability against exploiters specifically.

When analyzing a particular strategy 'i' three things are necessary to obtain in order to obtain its performance and stability.

1. $V(i,i)$.
2. The strategy 'j' that maximally exploits 'i'.
3. $V(j,i)$.

The first is needed to evaluate the performance, while the second and third are needed to evaluate stability. If strategy i is not stable, then both the T and R variables are increased to incentivize cooperation until either an absurdly high amount is reached, or until strategy i becomes stable. This is repeated for several different noise amounts.

3 Analysis of several strategies

In this paper, 6 strategies were analyzed. The first four strategies were analytically solved (3.1, 3.2, 3.3, and 3.4). For strategies 5 and 6 simulations were used (3.5 and 3.6). To obtain $V(i,i)$ and $V(j,i)$ through simulation a match between the two relevant strategies was simulated and matches last two million turns. In each section, the corresponding strategy is explained and it is then shown how the necessary three parts were obtained. The results will be shown in the results section.

3.1 TFT

To obtain $V(TFT, TFT)$ Markov chains are used. Inspiration was taken from [1, 11]. Let us denote cooperation as '0', defection as '1', and the amount of noise as n with $0 < n < 0.5$. Since TFT players only remember the opponents' last move, each TFT player can be in two possible states at all time. Therefore, two TFT players can be in four possible states: 1. (0, 0), 2. (0, 1), 3. (1, 0) and 4. (1, 1). Where (0, 0) means that both players cooperated on the last move. A 4x4 transition matrix A can be created, which describes the probabilities of each state transition. For example, A_{12} describes the probability of going from the state (0,0) to the state (0, 1). In this case:

$$A_{TFT,TFT} = \begin{pmatrix} (1-n)^2 & n(1-n) & n(1-n) & n^2 \\ n(1-n) & n^2 & (1-n)^2 & n(1-n) \\ n(1-n) & (1-n)^2 & n^2 & n(1-n) \\ n^2 & n(1-n) & n(1-n) & (1-n)^2 \end{pmatrix}$$

At the start of each match, both players have a clean sheet and therefore want to cooperate. The match is in the state of (0, 0). This can be denoted as $x_0 = (1, 0, 0, 0)$. To obtain the next state x_1, A must be multiplied by x_0. In general: $x_n = Ax_{n-1}$. The long-term distribution of the states can be obtained by: $\lim_{n\to\infty} A^n x_0$, which is the normalized eigenvector of the eigenvalue 1.

Having obtained the state distributions it is possible to calculate $V(TFT, TFT)$ by multiplying the frequencies of each state by the corresponding payoff. For example, in the state (0, 1) player 1 would receive S.

Claim: There does not exist a strategy j such that $V(j,i) > V(i,j)$. If true then TFT will be stable.

Without noise it is trivial. The exploiter must try to defect at one point to try and beat TFT. However, when trying to do so, it will receive a defect back. Both players receive $T + S$. Chaining defects will result in a payoff of P for both players. It is not possible to score higher or lower than TFT in an infinite game.

Proof with noise:

When the exploiter tries to defect for the first time, TFT would try to cooperate. Let us denote this as DC. When both players want to play DC, DC will happen with probability $(1-n)^2$ and CD with probablity n^2. Since $1 - n > n$, the exploiter is ahead, but only momentarily. If the exploiter after DC was played now wants to cooperate, CD is likely to happen. However, now everything is flipped and the exploiter cannot win. If the exploiter keeps defecting, DD is likely to happen. However here DC and CD, which are the only outcomes with a score difference are equally probable and from this point the exploiter cannot win. Therefore, in an infinite game, it is impossible to beat TFT.

3.2 TF2T

$V(TF2T, TF2T)$ is obtained in a similar fashion. The only difference is the transition matrix. Since $TF2T$ has a memory size of two, there are four different possible states for one player, and sixteen different possible states for two player, resulting in a 16x16 matrix. The first state is denoted as $(00, 00)$ where both players cooperated the last two turns. Each next state is logically follows the previous state as if counting in the binary system. For example the second state is denoted as $(00, 01)$ and the seventh state would be $(01, 10)$. The $TF2T$ transition matrix can be found in the appendix.

The strategy that maximally exploits $TF2T$ is the alternator strategy denoted as ALT. This strategy alternates between cooperate and defect. Note that this strategy does not cooperate on all even turns and defects on all odd turns. Instead it has a memory of size 1, and will play the opposite of what it played before. This crucial difference reduces the odds of defecting twice under noise.

Proof: Since $TF2T$ remembers the opponents' last two moves, the exploiting strategy only needs a memory size of one. Also, since noise is less than 0.5 and $2R > T + S$, defecting twice in a row must be avoided. ALT being the best exploiter logically follows. If memory is D then cooperating is best. If C, then defecting is best. Defecting at a later moment simply wastes time.

The matrix used to obtain $V(ALT, TF2T)$ will have eight states. For example state $(1, 01)$ is the fifth stage and means that ALT defected on the last turn and that $TF2T$ remembers that ALT cooperated two turns ago, but defected on the last turn.

$$A_{ALT,TF2T} = \begin{pmatrix} n(1-n) & n^2 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 \\ 0 & 0 & n^2 & n(1-n) & 0 & 0 & n(1-n) & (1-n)^2 \\ n(1-n) & n^2 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 \\ 0 & 0 & n^2 & n(1-n) & 0 & 0 & n(1-n) & (1-n)^2 \\ n(1-n) & n^2 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 \\ 0 & 0 & n^2 & n(1-n) & 0 & 0 & n(1-n) & (1-n)^2 \\ n(1-n) & n^2 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 \\ 0 & 0 & n^2 & n(1-n) & 0 & 0 & n(1-n) & (1-n)^2 \end{pmatrix}$$

3.3 TFNT with $1 < n < 2$

Since $TF2T$ was found to be very exploitable, the restriction $1 < n < 2$ applies. This means that this range of strategies lies between TFT and $TF2T$. The strategy, referred to as $TFNT$, works as follows. First, if the opponent defected twice in the last two turns, $TFNT$ would defect as well. Second, if the opponent defected the last turn, then $TFNT$ would defect with a probability of $2 - n$.

For example, $TF1.6T$ would want to defect 40% of the time if the opponent defected the last turn.

$V(TF2T, TF2T)$ is obtained by slightly altering the matrix that was used to obtain $V(TF2T, TF2T)$. The matrix can again be found in the appendix.

The strategy that maximally exploits $TFNT$ is again the alternator. The exploiter wants to sneak in defects, and since $TFNT$'s memory is only two, it would be sub-optimal to wait longer.

3.4 GTFT

Generous Tit for Tat (GTFT) is similar to TFT, except that it does not retaliate to a single defection some ratio of the time, usually 10%. In this paper the full range of the retaliation ratio r is considered, that is between 0 and 1. If $r = 1$, then this would be equal to the TFT strategy. If $r = 0$, then this would be equal to the cooperator strategy. To improve readability of the matrix, let the probability of defecting after the opponent defected be equal to $d = r(1-n) + (1-r)n$, and the probability of cooperating after the opponent defected be equal to $c = (1-r)(1-n) + rn$. Then $V(GTFT, GTFT)$ can be obtained by altering the TFT transition matrix:

$$A_{GTFT,GTFT} = \begin{pmatrix} (1-n)^2 & n(1-n) & n(1-n) & n^2 \\ c(1-n) & cn & d(1-n) & dn \\ c(1-n) & d(1-n) & cn & dn \\ c^2 & cd & cd & dd \end{pmatrix}$$

To illustrate the difference between $GTFT$ and TFT, let $r = 0.8$ and $n = 0.1$. The transition matrices would like as follows:

$$A_{TFT} = \begin{pmatrix} 0.81 & 0.09 & 0.09 & 0.01 \\ 0.09 & 0.01 & 0.81 & 0.09 \\ 0.09 & 0.81 & 0.01 & 0.09 \\ 0.01 & 0.09 & 0.09 & 0.81 \end{pmatrix}$$

$$A_{GTFT} = \begin{pmatrix} 0.81 & 0.09 & 0.09 & 0.01 \\ 0.234 & 0.026 & 0.666 & 0.074 \\ 0.234 & 0.666 & 0.026 & 0.074 \\ 0.0676 & 0.1924 & 0.1924 & 0.5476 \end{pmatrix}$$

As expected $GTFT$ shows a positive bias towards the (0, 0) state, and a negative bias towards the (1, 1) state, compared with TFT.

The strategy that maximally exploits $GTFT$ is the defector DEF, who always defects. Since $GTFT$ only has a memory size of 1, the exploiter does not need memory at all. If defecting works, it should be done all the time.

$V(DEF, GTFT)$ is obtained using the following matrix:

$$A_{DEF,GTFT} = \begin{pmatrix} n(1-n) & n^2 & (1-n)^2 & n(1-n) \\ n(1-n) & n^2 & (1-n)^2 & n(1-n) \\ nc & nd & (1-n)c & (1-n)d \\ nc & nd & (1-n)c & (1-n)d \end{pmatrix}$$

3.5 CTFT

As previously mentioned $CTFT$ is a strategy similar to TFT, but has a elegant way of dealing with noise. $CTFT$ works as follows: An individual is either in good standing or not. At the start, an individual is always in good standing, but after defecting when $CTFT$ thinks it should not have, the individual is not in good standing anymore. To get back in good standing it needs to cooperate the next turn. Next, $CTFT$ cooperates if it is not in good standing, or if the opponent is in good standing.

Without noise, $CTFT$ works exactly like TFT and is therefore not exploitable.

With noise, $CTFT$ is still not exploitable. Proof: The invader can either be in good standing or not. Case 1: The invader is not in good standing. Both the invader's and $CTFT$ actions will get flipped with equal probability. $CTFT$ wants to defect, since it is assumed the invader is not in good standing. If only the invader's action gets flipped the invader gets S. If only $CTFT$'s action gets flipped the invader would receive T. If both actions get flipped, the invader would receive R, and if no actions were flipped the invader would receive P. Since the payoff of S and T, which are the only unequal payoffs, happen with equal probability the invader cannot beat $CTFT$, and therefore cannot exploit it. Case 2: The invader is in good standing. Now the invader can likely successfully defect and receive T, however it would then not be in good standing anymore. If the invader then continues to defect, it would receive P, which certainly does not beat two $CTFT$ players. If the invader cooperates if it is not in good standing it must take the sucker's payoff to get back in good standing, but since $T + S < 2R$, no progress is made.

Simulation was used to obtain $V(CTFT, CTFT)$. Since there does not exist strategy i that can exploit $CTFT$, this step is skipped, just like with TFT.

3.6 HTF2T

Hard tit for two tats (HTF2T) is a strategy that defects if the opponent defected twice in a row in the last three turns. So it would defect is the opponent's history is for example DDC, but not if it was DCD. The reason that a strategy that defects if the opponent defected twice out the the last three turns in general is not considered, is because performance of this strategy is extremely poor. Since $HTF2T$ strategy has a memory depth of three, simulation was used to obtain $V(HTF2T, HTF2T)$. The strategy, that optimally exploits $HTF2T$ is ALT, since two defections in a row must be avoided. Again, to obtain $V(ALT, HTF2T)$, simulation was used.

4 Results

All results can be found in the appendix in table format. To graph performance $T = 3, R = 2, P = 1, S = 0$ were used as the payoff rules. As stated before, to increase cooperation incentive with the goal of protection against exploitation both the T and R variables are increased. When evaluating stability, there is a cooperation incentive threshold where the strategy will become stable. Figure 2 gives an explanation of how to interpret the figures about stability.

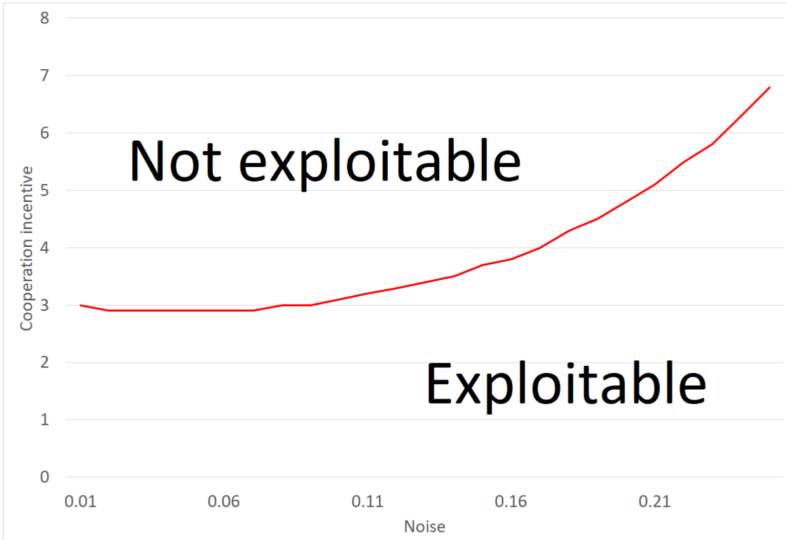

Figure 2: Threshold exploitability

Let us first look at the $TFNT$ strategies. First performance is compared and the stability. After $TFNT$, $GTFT$ is discussed, and after that the remaining strategies.

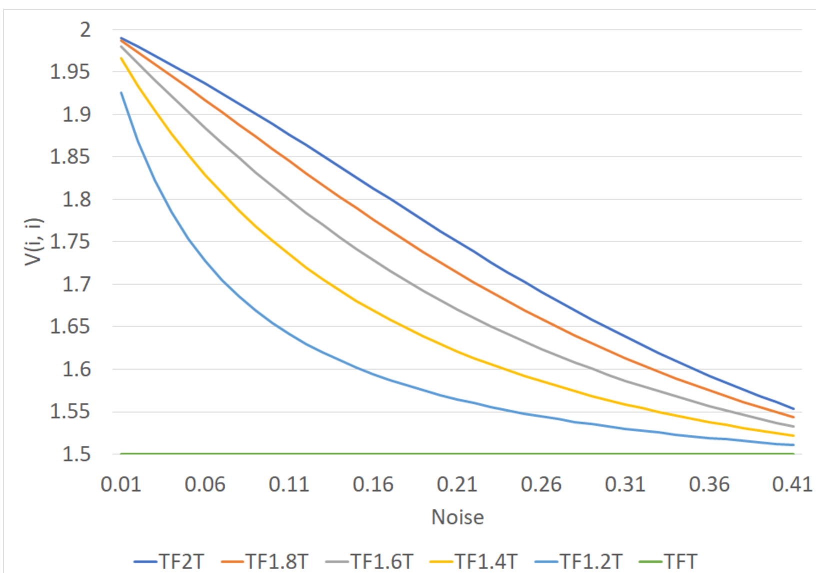

Figure 3: Performance comparison TFNT

As expected, performance goes up with more forgiveness. Notably, even a slight amount of forgiveness will have massive impact on performance, especially in environments with low amounts of noise.

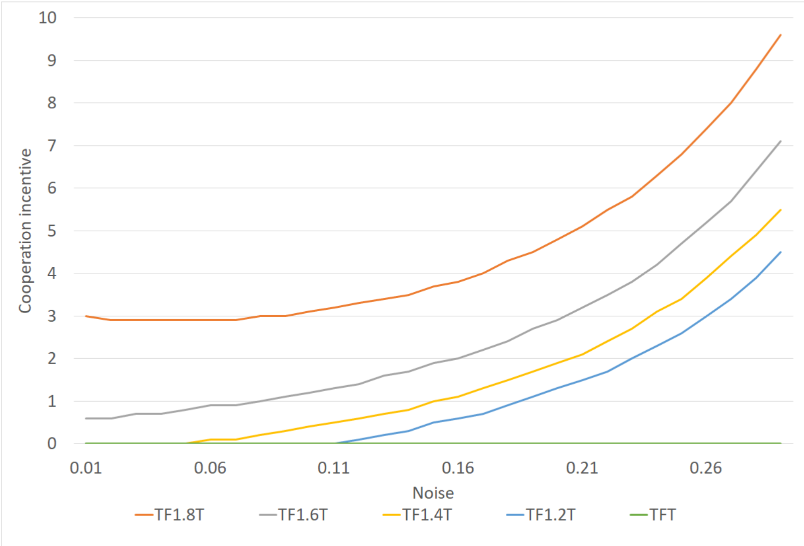

Figure 4: Stability comparison TFNT

While performance goes up, so does susceptibility to exploitation. As can be seen in figure 4, TFT is always stable, but strategies that show some forgiveness are in many cases still guaranteed to be stable. $TF2T$ was not included as it needed an extreme amount of help to become not exploitable.

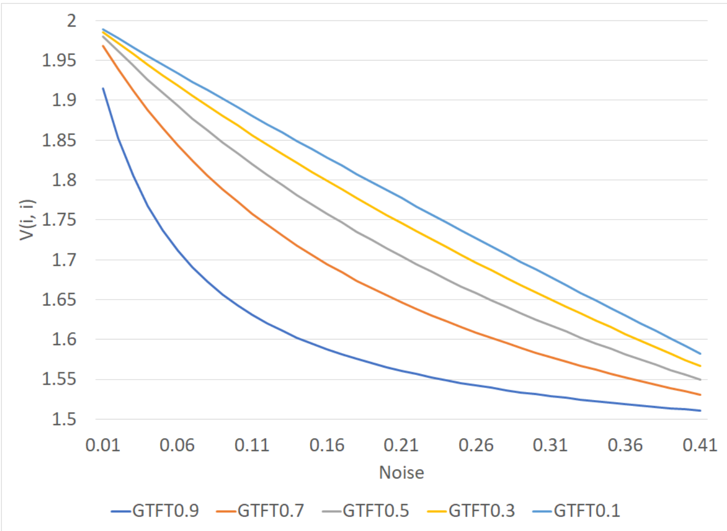

Figure 5: Performance comparison GTFT

Figure 5 looks extremely similar to figure 3. GTFT0.9 scores extremely well, compared to TFT, which would always score 1.5.

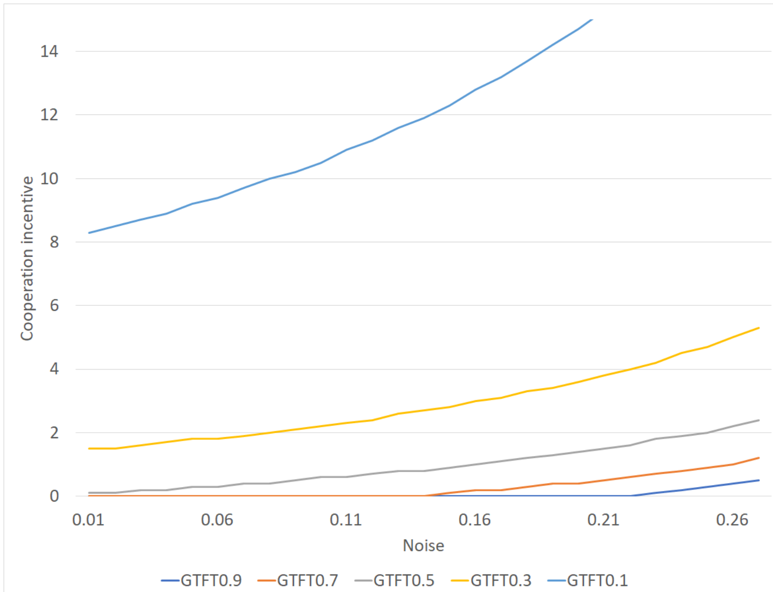

Figure 6: Stability comparison GTFT

Figure 6 again looks similar to figure 4. It is clear however that as noise increases susceptibility to exploitation increases less with $GTFT$ compared to $TFNT$. Figure 8 and 9 compares $GTFT$ and $TFNT$ directly.

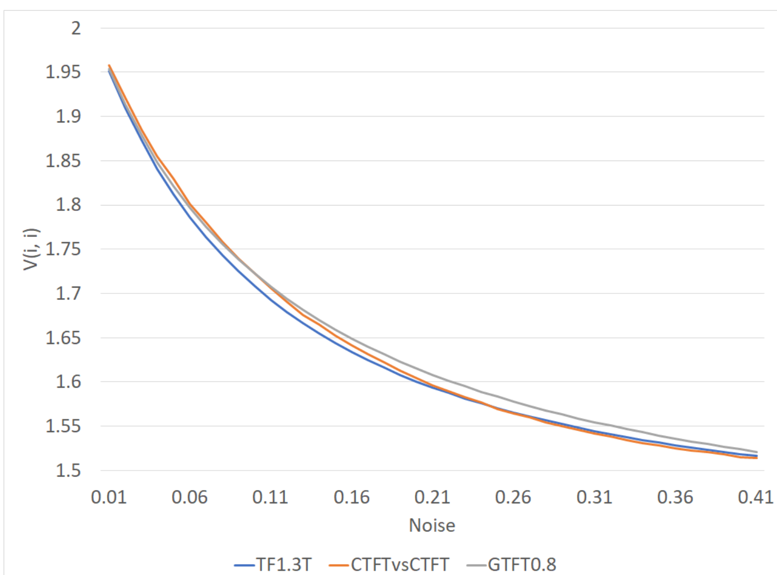

Figure 7: Performance comparison CTFT

$CTFT$ performs similarly to $TF1.3T$ and $GTFT0.8$. However, $CTFT$ is always stable, and is therefore preferred. Better performing than $TF1.3T$ and $GTFT0.8$ are $TFNT$ with $N > 1.3$ and $GTFTN$ with $N < 0.8$. If the environment has the right amount of noise and cooperation incentive that allows these strategies to be stable, the better performing strategies are of course preferred.

It was found that $HTF2T$ would never become stable and is therefore not included in any graphs. $HTF2T$'s performance can be found in the appendix.

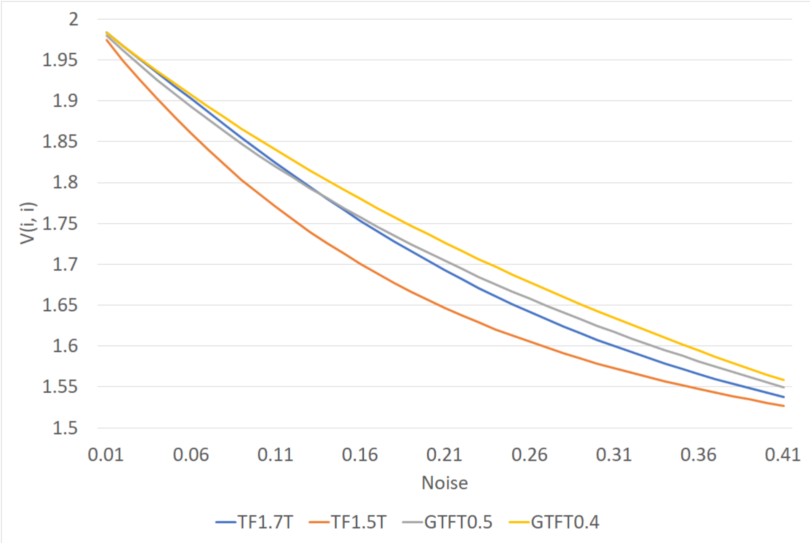

Figure 8: Performance comparison TFNT vs GTFT

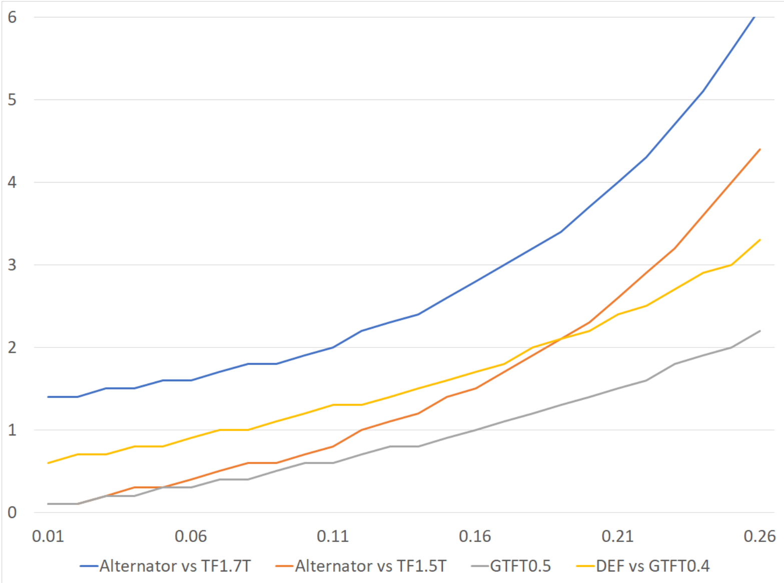

Figure 9: Stability comparison TFNT vs GTFT

Figure 8 shows that the performance of $GTFT0.4$ is greater than $TF1.7T$. Also $GTFT0.5$ performs better than $TF1.5T$. Figure 9 shows that in both cases $GTFT$, besides better performance, is also more stable.

More generally, $GTFT$ is always preferred over $TFNT$. More precisely put, for each strategy $TFNT$ with $1 < N < 2$, there exist a strategy $GTFT$ with retalation rate r, such that performance and stability is greater than that of $TFNT$ for all noise amounts. $GTFT$ is strictly preferred.

5 Conclusion and Future Research

In the context of performance and stability against exploitation, it was shown that some strategies are preferred over others in the iterated prisoner's dilemma. $GTFT$ performs both better and is more stable than $TFNT$, but for environments with a high noise amount and low cooperation incentive $CTFT$ will be the best performing stable strategy. However, stability as a whole was not considered in this paper. When a population cannot be invaded by exploiters, it can still be invaded by more cooperative strategies. Whereas exploiting strategies will strongly invade, cooperative strategies will only weakly invade the native population. What follows in the cooperative case is a heterogeneous environment with the original strategy and a more generous or forgiving strategy. What happens now with respect to susceptibility to exploitation is unclear and it is entirely possible that $TFNT$, while not preferred in the context of this paper, will perform and survive better in heterogeneous environments.

Furthermore, the strategies analyzed in this paper were only a few. While they were good candidates, because of their ability to deal with noise, it is possible there are alternative strategies that are more stable, perform better, or both.

6 References

References

[1] YJT (https://math.stackexchange.com/users/731237/yjt). *Cooperation percentage between 2 TF2T players (tit for two tats) in a noisy environement in the iterated prisoner's dilemma.* URL: `https://math.stackexchange.com/questions/4125846/cooperation-percentage-between-2-tf2t-players-tit-for-two-tats-in-a-noisy-envi`.

[2] Robert Axelrod. Basic Books, New York, 1984. ISBN: 0-465-02121-2.

[3] Robert Axelrod and Douglas Dion. "The further evolution of cooperation". In: *Science* (1988). DOI: `10.1126/science.242.4884.1385`.

[4] Robert Axelrod and William Hamilton. "The Evolution of Cooperation". In: *Science* (1981). DOI: `10.1126/science.7466396`.

[5] Jonathan Bendor. “In good times and bad: Reciprocity in an uncertain world”. In: *American Journal of Political Science* (1987). DOI: `https://doi.org/10.2307/2111282`.

[6] Jonathan Bendor and Piotr Swistak. “Types of evolutionary stability and the problem of cooperation”. In: *PNAS* (1995). DOI: `https://doi.org/10.1073/pnas.92.8.3596`.

[7] Robert Boyd. “Mistakes allow evolutionary stability in the repeated prisoner’s dilemma game”. In: *Science* (1989). DOI: `https://doi.org/10.1016/S0022-5193(89)80188-2`.

[8] Roderick Kramer Jonathan Bendor and Suzanne Stout. “When in Doubt... Cooperation in a Noisy Prisoner’s Dilemma”. In: *The Journal of Conflict Resolution* (1991). DOI: `https://doi.org/10.1177/0022002791035004007`.

[9] Per Molander. “The Optimal Level of Generosity in a Selfish, Uncertain Environment”. In: *The Journal of Conflict Resolution* (1985). DOI: `https://doi.org/10.1177/0022002785029004004`.

[10] Ulrich Mueller. “Optimal Retaliation for Optimal Cooperation”. In: *The Journal of Conflict Resolution* (1987). DOI: `https://doi.org/10.1177/0022002787031004008`.

[11] Martin Nowak and Karl Sigmund. “Invasion Dynamics of the Finitely Repeated Prisoner’s Dilemma”. In: *Games and Economic Behavior* (1994). DOI: `https://doi.org/10.1006/game.1995.1055`.

[12] Richard Wiederien and Firdaus Udwadia. “Global Patterns from Local Interactions: a Dynamical Systems Approach.” In: *International Journal of Bifurcation and Chaos* (2004). DOI: `https://doi.org/10.1142/S0218127404010953`.

[13] Jianzhong Wu and Robert Axelrod. “How to Cope with Noise in the Iterated Prisoner’s Dilemma”. In: *The Journal of Conflict Resolution* (1995). DOI: `https://doi.org/10.1177/0022002795039001008`.

7 Appendix

$$
A_{TF2T,TF2T} = \begin{matrix}
(1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
(1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & n(1-n) & n^2 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n(1-n) & n^2 & 0 & 0 & (1-n)^2 & n(1-n) \\
(1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
(1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & n(1-n) & n^2 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n(1-n) & (1-n)^2 & 0 & 0 & n^2 & n(1-n) & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n(1-n) & (1-n)^2 & 0 & 0 & n^2 & n(1-n) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n(1-n) & (1-n)^2 & 0 & 0 & n^2 & n(1-n) & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n^2 & n(1-n) & 0 & 0 & n(1-n) & (1-n)^2
\end{matrix}
$$

For the TFNT transition matrix, let $D = 2 - N$, which is the probability of wanting to defect after facing a single defect. Next defecting after wanting to defect $d = D(1-n) + (1-D)n$ and cooperating after wanting to defect $c = Dn + (1-D)(1-n)$. Then the TFNT transition matrix is equal to:

$$A_{TFNT,TFNT} = \begin{matrix}
(1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & c(1-n) & cn & 0 & 0 & d(1-n) & dn & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
(1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & n(1-n) & n^2 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c(1-n) & d(1-n) & 0 & 0 & cn & dn & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d^2 & cd & 0 & 0 & cd & c^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c(1-n) & d(1-n) & 0 & 0 & cn & dn & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & cn & dn & 0 & 0 & c(1-n) & d(1-n) \\
(1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & c(1-n) & cn & 0 & 0 & d(1-n) & dn & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
(1-n)^2 & n(1-n) & 0 & 0 & n(1-n) & n^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & n(1-n) & n^2 & 0 & 0 & (1-n)^2 & n(1-n) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n(1-n) & (1-n)^2 & 0 & 0 & n^2 & n(1-n) & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & cn & c(1-n) & 0 & 0 & dn & d(1-n) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n(1-n) & (1-n)^2 & 0 & 0 & n^2 & n(1-n) & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n^2 & n(1-n) & 0 & 0 & n(1-n) & (1-n)^2
\end{matrix}$$

Performance table:

noise	TF2T	TF1.9T	TF1.8T	TF1.7T	TF1.6T	TF1.5T	TF1.4T	TF1.3T	TF1.2T	TF1.1T	TFT	HTF2T	CTFT	GTFT0.9	GTFT0.8	GTFT0.7	GTFT0.6	GTFT0.5	GTFT0.4	GTFT0.3	GTFT0.2	GTFT0.1
0.01	1.99	1.99	1.99	1.98	1.98	1.97	1.97	1.95	1.93	1.86	1.50	1.97	1.96	1.92	1.95	1.97	1.98	1.98	1.98	1.99	1.99	1.99
0.02	1.98	1.98	1.97	1.97	1.96	1.95	1.93	1.91	1.87	1.78	1.50	1.95	1.92	1.85	1.91	1.94	1.95	1.96	1.97	1.97	1.98	1.98
0.03	1.97	1.96	1.96	1.95	1.94	1.93	1.90	1.87	1.82	1.73	1.50	1.92	1.89	1.81	1.88	1.91	1.93	1.94	1.95	1.96	1.96	1.97
0.04	1.96	1.95	1.95	1.94	1.92	1.90	1.88	1.84	1.78	1.69	1.50	1.90	1.86	1.77	1.85	1.89	1.91	1.93	1.94	1.94	1.95	1.96
0.05	1.95	1.94	1.93	1.92	1.90	1.88	1.85	1.81	1.75	1.66	1.50	1.88	1.83	1.74	1.82	1.86	1.89	1.91	1.92	1.93	1.94	1.95
0.06	1.94	1.93	1.92	1.90	1.88	1.86	1.83	1.79	1.73	1.64	1.50	1.86	1.80	1.71	1.80	1.84	1.87	1.89	1.91	1.92	1.93	1.93
0.07	1.92	1.91	1.90	1.89	1.87	1.84	1.81	1.76	1.71	1.62	1.50	1.84	1.78	1.69	1.78	1.82	1.86	1.88	1.89	1.91	1.92	1.92
0.08	1.91	1.90	1.89	1.87	1.85	1.82	1.79	1.74	1.69	1.61	1.50	1.82	1.76	1.67	1.76	1.81	1.84	1.86	1.88	1.89	1.90	1.91
0.09	1.90	1.89	1.87	1.85	1.83	1.80	1.77	1.72	1.67	1.60	1.50	1.80	1.74	1.66	1.74	1.79	1.82	1.85	1.87	1.88	1.89	1.90
0.10	1.89	1.88	1.86	1.84	1.82	1.79	1.75	1.71	1.65	1.59	1.50	1.79	1.72	1.64	1.72	1.77	1.81	1.83	1.85	1.87	1.88	1.89
0.11	1.88	1.86	1.85	1.82	1.80	1.77	1.74	1.69	1.64	1.58	1.50	1.77	1.71	1.63	1.71	1.76	1.79	1.82	1.84	1.86	1.87	1.88
0.12	1.86	1.85	1.83	1.81	1.78	1.75	1.72	1.68	1.63	1.57	1.50	1.76	1.69	1.62	1.69	1.74	1.78	1.81	1.83	1.84	1.86	1.87
0.13	1.85	1.84	1.82	1.80	1.77	1.74	1.71	1.67	1.62	1.57	1.50	1.74	1.68	1.61	1.68	1.73	1.77	1.79	1.82	1.83	1.85	1.86
0.14	1.84	1.82	1.80	1.78	1.76	1.73	1.69	1.65	1.61	1.56	1.50	1.73	1.66	1.60	1.67	1.72	1.75	1.78	1.80	1.82	1.84	1.85
0.15	1.83	1.81	1.79	1.77	1.74	1.71	1.68	1.64	1.60	1.55	1.50	1.72	1.65	1.59	1.66	1.71	1.74	1.77	1.79	1.81	1.83	1.84
0.16	1.81	1.80	1.78	1.75	1.73	1.70	1.67	1.63	1.59	1.55	1.50	1.70	1.64	1.59	1.65	1.69	1.73	1.76	1.78	1.80	1.81	1.83
0.17	1.80	1.78	1.76	1.74	1.72	1.69	1.66	1.62	1.59	1.55	1.50	1.69	1.63	1.58	1.64	1.68	1.72	1.75	1.77	1.79	1.80	1.82
0.18	1.79	1.77	1.75	1.73	1.70	1.68	1.65	1.62	1.58	1.54	1.50	1.68	1.62	1.58	1.63	1.67	1.71	1.74	1.76	1.78	1.79	1.81
0.19	1.78	1.76	1.74	1.72	1.69	1.67	1.64	1.61	1.58	1.54	1.50	1.67	1.61	1.57	1.62	1.66	1.70	1.72	1.75	1.77	1.78	1.80
0.20	1.76	1.74	1.73	1.70	1.68	1.66	1.63	1.60	1.57	1.54	1.50	1.66	1.60	1.57	1.62	1.66	1.69	1.71	1.74	1.76	1.77	1.79
0.21	1.75	1.73	1.71	1.69	1.67	1.65	1.62	1.59	1.56	1.53	1.50	1.65	1.60	1.56	1.61	1.65	1.68	1.70	1.73	1.75	1.76	1.78
0.22	1.74	1.72	1.70	1.68	1.66	1.64	1.61	1.59	1.56	1.53	1.50	1.64	1.59	1.56	1.60	1.64	1.67	1.69	1.72	1.74	1.75	1.77
0.23	1.73	1.71	1.69	1.67	1.65	1.63	1.61	1.58	1.56	1.53	1.50	1.63	1.58	1.55	1.60	1.63	1.66	1.68	1.71	1.73	1.74	1.76
0.24	1.71	1.70	1.68	1.66	1.64	1.62	1.60	1.58	1.55	1.53	1.50	1.63	1.58	1.55	1.59	1.62	1.65	1.68	1.70	1.72	1.73	1.75
0.25	1.70	1.69	1.67	1.65	1.63	1.61	1.59	1.57	1.55	1.52	1.50	1.62	1.57	1.55	1.58	1.62	1.64	1.67	1.69	1.71	1.72	1.74
0.26	1.69	1.68	1.66	1.64	1.62	1.61	1.59	1.57	1.54	1.52	1.50	1.61	1.56	1.54	1.58	1.61	1.63	1.66	1.68	1.70	1.71	1.73
0.27	1.68	1.67	1.65	1.63	1.62	1.60	1.58	1.56	1.54	1.52	1.50	1.60	1.56	1.54	1.57	1.60	1.63	1.65	1.67	1.69	1.70	1.72
0.28	1.67	1.65	1.64	1.62	1.61	1.59	1.57	1.56	1.54	1.52	1.50	1.60	1.55	1.54	1.57	1.60	1.62	1.64	1.66	1.68	1.69	1.71
0.29	1.66	1.64	1.63	1.62	1.60	1.58	1.57	1.55	1.54	1.52	1.50	1.59	1.55	1.53	1.56	1.59	1.61	1.63	1.65	1.67	1.68	1.70
0.30	1.65	1.64	1.62	1.61	1.59	1.58	1.56	1.55	1.53	1.52	1.50	1.58	1.55	1.53	1.56	1.58	1.61	1.63	1.64	1.66	1.67	1.69
0.31	1.64	1.63	1.61	1.60	1.59	1.57	1.56	1.54	1.53	1.52	1.50	1.58	1.54	1.53	1.55	1.58	1.60	1.62	1.63	1.65	1.66	1.68
0.32	1.63	1.62	1.60	1.59	1.58	1.57	1.55	1.54	1.53	1.51	1.50	1.57	1.54	1.53	1.55	1.57	1.59	1.61	1.63	1.64	1.66	1.67
0.33	1.62	1.61	1.60	1.59	1.57	1.56	1.55	1.54	1.53	1.51	1.50	1.56	1.53	1.52	1.55	1.57	1.59	1.60	1.62	1.63	1.65	1.66
0.34	1.61	1.60	1.59	1.58	1.57	1.56	1.55	1.53	1.52	1.51	1.50	1.56	1.53	1.52	1.54	1.56	1.58	1.60	1.61	1.62	1.64	1.65
0.35	1.60	1.59	1.58	1.57	1.56	1.55	1.54	1.53	1.52	1.51	1.50	1.55	1.53	1.52	1.54	1.56	1.57	1.59	1.60	1.62	1.63	1.64
0.36	1.59	1.58	1.57	1.57	1.56	1.55	1.54	1.53	1.52	1.51	1.50	1.55	1.53	1.52	1.54	1.55	1.57	1.58	1.59	1.61	1.62	1.63
0.37	1.58	1.58	1.57	1.56	1.55	1.54	1.53	1.53	1.52	1.51	1.50	1.55	1.52	1.52	1.53	1.55	1.56	1.57	1.59	1.60	1.61	1.62
0.38	1.58	1.57	1.56	1.55	1.55	1.54	1.53	1.52	1.52	1.51	1.50	1.54	1.52	1.52	1.53	1.54	1.56	1.57	1.58	1.59	1.60	1.61
0.39	1.57	1.56	1.56	1.55	1.54	1.53	1.53	1.52	1.51	1.51	1.50	1.54	1.52	1.51	1.53	1.54	1.55	1.56	1.57	1.58	1.59	1.60
0.40	1.56	1.56	1.55	1.54	1.54	1.53	1.52	1.52	1.51	1.51	1.50	1.53	1.51	1.51	1.52	1.53	1.55	1.56	1.57	1.57	1.58	1.59
0.41	1.55	1.55	1.54	1.54	1.53	1.53	1.52	1.52	1.51	1.51	1.50	1.53	1.51	1.51	1.52	1.53	1.54	1.55	1.56	1.57	1.57	1.58

Stability table:

noise	TF2T	TF1.9T	TF1.8T	TF1.7T	TF1.6T	TF1.5T	TF1.4T	TF1.3T	TF1.2T	TF1.1T	GTFT0.9	GTFT0.8	GTFT0.7	GTFT0.6	GTFT0.5	GTFT0.4	GTFT0.3	GTFT0.2	GTFT0.1
0.01	100.1	7.4	3	1.4	0.6	0.1	0	0	0	0	0	0	0	0	0.1	0.6	1.5	3.2	8.3
0.02	50.3	7	2.9	1.4	0.6	0.1	0	0	0	0	0	0	0	0	0.1	0.7	1.5	3.3	8.5
0.03	33.7	6.6	2.9	1.5	0.7	0.2	0	0	0	0	0	0	0	0	0.2	0.7	1.6	3.4	8.7
0.04	25.6	6.2	2.9	1.5	0.7	0.3	0	0	0	0	0	0	0	0	0.2	0.8	1.7	3.5	8.9
0.05	20.7	6	2.9	1.6	0.8	0.3	0	0	0	0	0	0	0	0	0.3	0.8	1.8	3.6	9.2
0.06	17.5	5.8	2.9	1.6	0.9	0.4	0.1	0	0	0	0	0	0	0	0.3	0.9	1.8	3.7	9.4
0.07	15.3	5.6	2.9	1.7	0.9	0.5	0.1	0	0	0	0	0	0	0	0.4	1	1.9	3.9	9.7
0.08	13.7	5.5	3	1.8	1	0.6	0.2	0	0	0	0	0	0	0	0.4	1	2	4	10
0.09	12.5	5.4	3	1.8	1.1	0.6	0.3	0	0	0	0	0	0	0.1	0.5	1.1	2.1	4.1	10.2
0.1	11.6	5.3	3.1	1.9	1.2	0.7	0.4	0.1	0	0	0	0	0	0.1	0.6	1.2	2.2	4.3	10.5
0.11	10.9	5.3	3.2	2	1.3	0.8	0.5	0.2	0	0	0	0	0	0.2	0.6	1.3	2.3	4.5	10.9
0.12	10.4	5.3	3.3	2.2	1.4	1	0.6	0.3	0.1	0	0	0	0	0.2	0.7	1.3	2.4	4.6	11.2
0.13	10.0	5.4	3.4	2.3	1.6	1.1	0.7	0.4	0.2	0	0	0	0	0.3	0.8	1.4	2.6	4.8	11.6
0.14	9.7	5.4	3.5	2.4	1.7	1.2	0.8	0.6	0.3	0.1	0	0	0	0.4	0.8	1.5	2.7	5	11.9
0.15	9.5	5.5	3.7	2.6	1.9	1.4	1	0.7	0.5	0.3	0	0	0.1	0.4	0.9	1.6	2.8	5.2	12.3
0.16	9.4	5.7	3.8	2.8	2	1.5	1.1	0.8	0.6	0.4	0	0	0.2	0.5	1	1.7	3	5.4	12.8
0.17	9.3	5.8	4	3	2.2	1.7	1.3	1	0.7	0.5	0	0	0.2	0.6	1.1	1.8	3.1	5.6	13.2
0.18	9.3	6	4.3	3.2	2.4	1.9	1.5	1.2	0.9	0.7	0	0	0.3	0.7	1.2	2	3.3	5.9	13.7
0.19	9.4	6.3	4.5	3.4	2.7	2.1	1.7	1.3	1.1	0.8	0	0.1	0.4	0.7	1.3	2.1	3.4	6.1	14.2
0.2	9.6	6.5	4.8	3.7	2.9	2.3	1.9	1.6	1.3	1	0	0.1	0.4	0.8	1.4	2.2	3.6	6.4	14.7
0.21	9.8	6.8	5.1	4	3.2	2.6	2.1	1.8	1.5	1.2	0	0.2	0.5	0.9	1.5	2.4	3.8	6.7	15.3
0.22	10.1	7.2	5.5	4.3	3.5	2.9	2.4	2	1.7	1.5	0	0.3	0.6	1	1.6	2.5	4	7	15.9
0.23	10.4	7.6	5.8	4.7	3.8	3.2	2.7	2.3	2	1.7	0.1	0.4	0.7	1.1	1.8	2.7	4.2	7.3	16.6
0.24	10.9	8.1	6.3	5.1	4.2	3.6	3.1	2.6	2.3	2	0.2	0.5	0.8	1.3	1.9	2.9	4.5	7.7	17.3
0.25	11.4	8.6	6.8	5.6	4.7	4	3.4	3	2.6	2.3	0.3	0.6	0.9	1.4	2	3	4.7	8.1	18
0.26	12.0	9.2	7.4	6.1	5.2	4.4	3.9	3.4	3	2.7	0.4	0.7	1	1.5	2.2	3.3	5	8.5	18.9
0.27	12.8	9.9	8	6.7	5.7	5	4.4	3.9	3.4	3.1	0.5	0.8	1.2	1.7	2.4	3.5	5.3	8.9	19.8
0.28	13.6	10.7	8.8	7.4	6.4	5.6	4.9	4.4	3.9	3.5	0.6	0.9	1.3	1.8	2.6	3.7	5.6	9.4	20.8
0.29	14.6	11.7	9.6	8.2	7.1	6.2	5.5	5	4.5	4.1	0.7	1	1.5	2	2.8	4	6	10	21.9
0.3	15.8	12.7	10.6	9.1	7.9	7	6.3	5.7	5.1	4.7	0.8	1.2	1.6	2.2	3.1	4.3	6.4	10.6	23.1
0.31	17.2	14	11.8	10.2	8.9	7.9	7.1	6.5	5.9	5.4	1	1.3	1.8	2.4	3.3	4.6	6.8	11.2	24.4
0.32	18.8	15.5	13.2	11.4	10.1	9	8.1	7.4	6.8	6.2	1.1	1.5	2	2.7	3.6	5	7.3	11.9	25.8
0.33	20.7	17.3	14.8	12.9	11.4	10.3	9.3	8.5	7.8	7.2	1.3	1.7	2.3	3	3.9	5.4	7.9	12.8	27.5
0.34	23.1	19.4	16.7	14.6	13	11.8	10.7	9.8	9	8.4	1.5	2	2.5	3.3	4.3	5.9	8.5	13.7	29.3
0.35	25.9	21.9	19	16.8	15	13.6	12.4	11.4	10.5	9.8	1.8	2.2	2.8	3.6	4.7	6.4	9.2	14.7	31.4
0.36	29.3	25	21.8	19.3	17.4	15.8	14.5	13.3	12.4	11.5	2	2.5	3.2	4	5.2	7	10	15.9	33.8
0.37	33.6	28.9	25.3	22.6	20.3	18.5	17	15.7	14.6	13.7	2.3	2.9	3.5	4.5	5.7	7.7	10.9	17.3	36.5
0.38	39.0	33.7	29.7	26.6	24.1	22	20.2	18.8	17.5	16.4	2.7	3.3	4	5	6.4	8.5	11.9	18.9	39.7
0.39	45.9	40	35.4	31.8	28.8	26.4	24.4	22.7	21.2	19.9	3.1	3.7	4.5	5.6	7.1	9.4	13.2	20.8	43.5
0.4	55.0	48.1	42.8	38.6	35.1	32.3	29.8	27.8	26	24.4	3.6	4.3	5.2	6.4	8	10.6	14.7	23	48.1
0.41	67.2	59.1	52.8	47.7	43.6	40.1	37.2	34.7	32.5	30.6	4.2	5	6	7.3	9.2	11.9	16.6	25.8	53.6