

Research of Parameters in Strategies of the Iterated Continuous Prisoner's Dilemma

Adrian Hossner

Contents

1	Introduction	3
2	Theoretical Foundations	3
3	Methods and Implementation	6
4	Results	9
5	Discussion	16
6	Conclusions and Outlook	20
7	Self Reflection	20
8	Annex	20

Abstract

1 Introduction

- Interactions
implication range is wide:
from two persons in a relationship to countries in alliance
from two impala's allogrooming to zebras warning the herd about a lion
- Key hook:
helping someone with possible cost – cooperation
being selfish but breaking trust – defection
- model PD:
single interaction
already well analysed
Nash found solution/equilibrium
- Iterated:
more interesting, insightful
strategies
- non-suited variants:
Axelrod's Tournament
Evolution
- continuous:
more complex, accurate
- parameters:
determines behaviour
create surface
the part that has not been experimented
- overall and difference:
overall: wealth of population
difference: competence between individuals

2 Theoretical Foundations

- Game Theory (optional, not needed for understanding)
mathematical framework to investigate interactions
- Prisoner's Dilemma:

The Prisoner's Dilemma (PD) is a famous intellectual game in Game Theory. It was invented by Merrill Flood and Melvin Dresher in 1950. In order to understand the dilemma, a proper context has to be established. Two persons are arrested and are sued for having stolen something. The police, however, has not enough evidence to imprison them. As a consequence of that, the police interrogates the two criminals. They can either say nothing or confess that the other suspect was involved in the crime. They can either cooperate or defect without having the possibility to talk

to each other before the decision. A certain tactic is being used by the police officers. They make it profitable for the criminals to confess. They let the criminal out of prison earlier if he/she confesses. But if the other also confesses, both stay long in prison. If he/she says nothing and the other confesses, he/she will stay very long in prison. If, however, both stay silent, both of them go to prison in a relatively short amount of time. The following matrix visualises the pay-off's in this game in a comprehensive way:

	C	D
C	3, 3	10, 0
D	0, 10	5, 5

C stands for cooperation and D stands for defection. Cooperation is defined by remaining silent whereas defection means confessing. The dilemma consists of the following. Confessing seems attractive since the interrogated criminal can walk out freely without going to prison. However, if the second criminal also confesses, both get five years in prison. This is, nevertheless, the worst outcome for both which could have been avoided as both could have stayed silent.

John Nash, a mathematician which made great contributions in Game Theory, has proved that it is the most logical option for both to confess, always. A Nash-equilibrium is introduced. It defines a stable state in which both would not change their decision even though they would know if the other cooperated or defected.

This game composes a one-time interaction very well in a mathematical and analytical perspective. The pay-off's in the matrix can be changed as long as these rules stay fulfilled.

	C	D
C	R, R	S, T
D	T, S	P, P

$$T > R > P > S$$

- Iterated:

The Iterated Prisoner's Dilemma (IPD) is a game which extends the PD. As the name suggests, the PD is played a number of times sequentially. After each round the pay-off gets accumulated to the points one strategy has already gained. A strategy is to be considered as an autonomous player which has its own behaviour. A strategy has the whole history of its own contributions and the ones of the opponent at hand to generate the decision for the next round. These decisions are calculated by using conditions and probabilities that will be applied on the given data. In this, and in the following variants, the analogy to the real world is changed. It is wanted to maximize their output in the end. In contrary to the initial variant, it means that high number are wanted instead of unwanted. To still hold on the idea of the Prisoner's Dilemma, the pay-off's could be interpreted as to be subtracted. Meaning, the bigger the points, the sooner one gets out of prison. Since a strategy wants to gain as many points as possible, the pay-off matrix should be altered accordingly.

	C	D
C	3, 3	0, 5
D	5, 0	1, 1

And to generalise this matrix as well, the numbers can be replaced with variables. These variables T, R, P and S

	C	D
C	R, R	S, T
D	T, S	P, P

$$T < R < P < S$$

$$2R > T + S$$

The Nash-equilibrium in the IPD differs from that in the PD. This means that cooperation can earn out long-term. By cooperating, one is building trust with another. One strategy can only get the maximum points, which can be gained by always getting the 5 points each round, by defecting and the opponent's cooperation. Since the opponent will likely not cooperate after one strategy always defects, the maximum is practically impossible to receive. So, the relative high amount and the possibility of getting it for both is by cooperating.

The number of round that will be played has to be unknown to both strategies. There is no sense in cooperating the last round since trust does not matter after the game is ended. And because trust will be given up in the last round, there is no sense for both to keep trust the second last round. This will end up in an inductive defection behaviour for both strategies the whole game.

This extension of the PD allows us to simulate a relationship over time where the two players interact multiple times.

- Continuous:

The continuous variant is considered more accurate to the real world since the decisions are rather rare only either cooperation or defection. The amount of cooperation is called investment and can range from zero to one (0 to 1) where full defection is represented by 0 and cooperation corresponds to 1. One would suspect a very similar pay-off system to the iterated variant. In the discrete version, it looked like this:

Pay-off Player A			Pay-off Player B		
	C_A	D_A		C_A	D_A
C_B	3	5	C_B	3	0
D_B	0	1	D_B	5	1

This is possible by bilinearly interpolating the pay-off matrix so that the lines become fluid. This will establish a field that can be represented by colour grades, a heat map. The equation for utilizing this heat map correctly are also displayed.

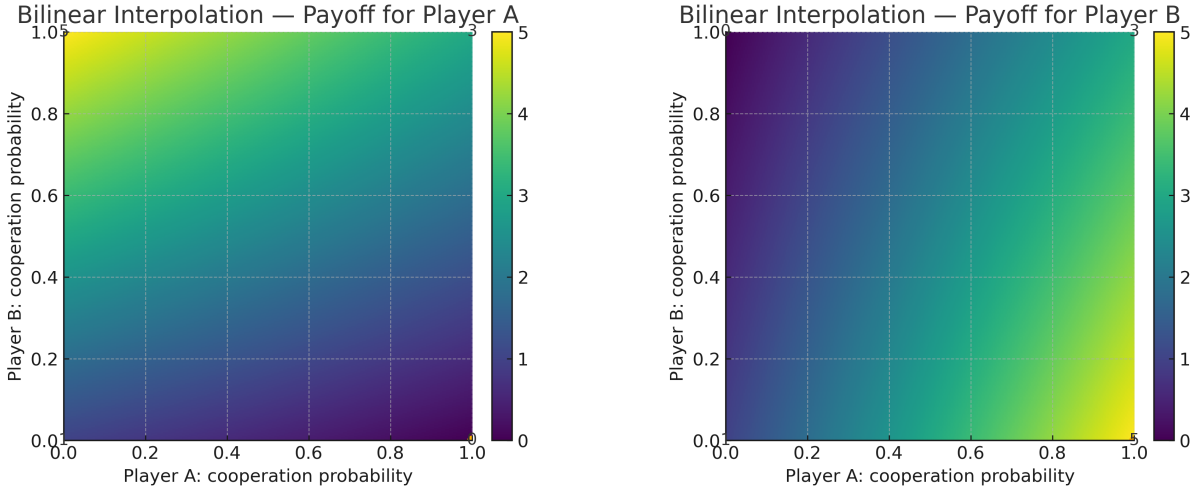


Figure 1: Pay-off hash maps in the continuous variant of the IPD.

I used ChatGPT-o4 to generate these heat maps. It generated them using these functions:

$$U_A = (1 - x)(1 - y)R_A + (1 - x)yS_A + x(1 - y)T_A + xyP_A$$

$$U_B = (1 - x)(1 - y)R_B + (1 - x)yS_B + x(1 - y)T_B + xyP_B$$

These functions are applied to every point on the map. The variable x is the investment of strategy A and y is the investment of strategy B. There is, however, another way of determine the pay-off's.

This approach uses only equations instead. It is much easier to implement but understanding is the difficult part. This pay-off system of [sciencedirect.com] is formed by:

$$P_A = y - c * x + c$$

$$P_B = x - c * y + c$$

+ c because no negative numbers. P_A and P_B are the pay-off's of strategy A and B respectively. The coefficient c ranges, as well as x and y, from 0 to 1 and describes the cost-to-benefit ratio. The base of the pay-off's are the opponent's investment. Then the strategy's own investment gets multiplied by the coefficient c and then is subtracted from the base. This means the pay-off is highly influenced by the opponent's investment. Let's see if the principles of the initial PD are still valid. It is better for one to defect (submit a 0) to get rid of the subtraction. And it is better for the strategy if the opponent's cooperates. If both defect, one gets zero points. If both cooperate, both get $1 - c$ points. If, however, one strategy exploits the opponent, meaning it defects while the other cooperates, it will get the maximum points being 1. The opponent would then get $-c$ points. This is the least one can receive. [Examples of coefficient c]

- Noise:

Noise in the ICPD is the equivalent to miscommunication or misunderstanding in the real world. In this variant of the PD, noise is essential to trigger interesting outcomes. When a game of the ICPD is started and two strategies who start with a full investment, many strategies only respond with full investments. So noise is necessary to trigger continuous investments. Without noise, in many cases, the ICPD would be equivalent to the IPD.

- Simultaneous vs Alternating:

Two main differences can be seen when simulating an iterated variant of the PD. On one hand there is the simultaneous form whereby the strategies submit their contribution at the same time. On the other hand, there is the alternating form in which one strategy submits its contribution and the other strategy can respond to this contribution in that round. Since the alternating form gives an advantage to the strategy which is allowed to respond, this only makes things more complicated than necessary. So, this analysis is only dedicated to the simultaneous form.

- Current Findings:

In the simpler variants, e.g. the IPD without noise, it has been proved many times over that the strategy Tit-For-Tat is the most successful one. [sources] The success of strategies, however, can vary depending on certain conditions such as the influence of noise or different pay-off systems where continuous investments can be submitted. So, the very specific area of the simultaneous ICPD, has not been very well explored. ...

3 Methods and Implementation

- Parameter-based:

The main idea of this paper is to use parameter-based strategies. These strategies hold a parameter that defines the behaviour of the strategy. The parameter will always range from 1 to 10 inclusively. [Example]

- Surfaces:

The end result of this paper will be several surface plots. The data will be generated by letting two parameter-based strategies play against each other in the ICPD. The game will be structured so that every parameter came against every other parameter in the ICPD. In total, I will let them play the ICPD 100 times in order to smooth out abnormalities which happen only once. Like this,

a surface can be plotted by having the x-axis being the parameter of strategy 1 and the y-axis being the parameter of strategy 2. The z-axis will indicate the points one strategy gained. Since there are two strategies in one game, one surface will be shown for each strategy. Further more, the two z-axis of the two surfaces can be added and form a new surface which shows the overall points gained by both strategies. The complement to that would be to subtract the two z-axis to plot a surface showing the difference between the two surfaces. This describes how much better one strategy was than the other. So, in the end there will be four surface plots per one ICPD.

- Simulated Strategies and explanation of parameter:

My selection:

In this simulation, strategies have to be defined and selected. There are two main points I want to investigate in this project. First, I want to associate the strategies to specific groups that differ in the characteristic of each member. I suggest the following three characteristics:

1. Responsive
2. Rigid
3. Random

I chose these three qualities due to the fact that one has to follow one of these three natures. In the real world, every personality can be associated with one of my defined groups. It can also be looked at this from another perspective. Whenever one has to respond to another's action, one has to decide how to answer. The shift applied to the previous investment can be categorised in one of these three groups. Either the shift is conditional to the opponent's last contribution, then this strategy is responsive. Or the shift is always 0 then the behaviour falls into the group of the rigid. This means that the investment will unconditionally stay the same. Or the shift is neither one of the above, meaning it doesn't depend on the past interactions nor stays it always 0. Then there cannot any determining aspect be established, meaning it is random.

Second, the question arises if there is a difference between the ICPD and the IPD. So, I propose a discrete and a continuous variant of each group. With the exception of the group 'Rigid', there will be two members of each group. It will be clarified why only one strategy in this group makes sense. Thus, we can analyse the difference between the two members of the group to clarify which variant gained more points in the ICPD. Having two strategies in each group except in the group 'Rigid', means implementing five strategies. And letting all strategies play against every other strategy, including itself, means having enough data to plot $4 \cdot \sum_{a=1}^5$ surfaces. In total, this would equal to 60 surface plots.

The following strategies are described by algorithmising them. $i(\theta)$ is a function that corresponds to the investment in the current round and holds the variable θ which is the parameter of the strategy in this round. I showcase what the behaviour of the strategy will be equivalent to when the parameter is 0, 5 and 10 for an easier understanding.

Adapt-Continuous:

Adapt-Continuous (AdpC) is an, as the name suggests, adaptive and thus responsive strategy. It starts with full cooperation to offer a constructive relationship within this simulation. After having played one round, it will adapt to the submissions of the opponent by shifting its own investment towards the opponent's.

$$i(\theta_{\text{AdpC}}) = i_1 + s(\theta_{\text{AdpC}})$$

$$s(\theta_{\text{AdpC}}) = \frac{\theta_{\text{AdpC}}}{5} \cdot (\bar{i}_1 - i_1)$$

The shift being applied to its own previous investment is notated with the function $s(\theta_{\text{AdpC}})$. This strategy with the parameter 0 is identical to AlwaysCooperate since the difference and thus the shift is multiplied by 0. Dividing θ_{AdpC} by 5 implies the fact that this coefficient to the difference is equal to one if the parameter equals five. This means that the strategy will shift its next investment to exactly the previous investment of the opponent. This behaviour corresponds to Tit-For-Tat. $\theta_{\text{AdpC}} = 10$ also shows an interesting nature of the strategy. By having the parameter set to ten, it means Adapt will add the shift to its own previous investment twice. Of course, one strategy is not allowed to submit any investment that exceeds the limits that is defined to be from 0 to 1. The implementation of this strategy will simply set its investment to the reached limit if surpassed.

Adapt-Discrete:

As I mentioned, there is one continuous and one discrete strategy. This means there must be an Adapt-Discrete (AdpD) implemented. The simplest implementation of this strategy is to take the same investment function as Adapt-Continuous and round the result.

$$i(\theta_{\text{AdpD}}) = \lfloor i_1 + s(\theta_{\text{AdpD}}) \rfloor$$

$$s(\theta_{\text{AdpD}}) = \frac{\theta_{\text{AdpD}}}{5} \cdot (\bar{i}_1 - i_1)$$

Something that is not seen in these equations is that the strategy will start with full cooperation, as Adapt-Continuous. If $i(\theta_{\text{AdpD}}) \geq 0.5$, the ultimate investment will be 1. Otherwise it will be equal to 0. Once the strategy is at full defection, it can only get to cooperation again if $\theta_{\text{AdpD}} \geq 3$. Here is why:

The maximum difference there can be is when this strategy submits 1 and the opponent submits a 0, so 1. Looking at the function, this threshold of 0.5 can only be overstepped if the shift is at least equal to 0.5. This only can be achieved if the coefficient is greater or equal to 0.5 ($\frac{\theta_{\text{AdpD}}}{5} \geq 0.5$). This inequation is fulfilled if $\theta_{\text{AdpD}} \geq 3$.

$\theta_{\text{AdpD}} = 0$ means, as in the continuous version, that the strategy corresponds to AlwaysCooperate because the shift is equal to 0 and thus always stays at its first investment, being full cooperation. If the parameter is equal to 10, it is most likely to jump from 1 to 0 back to 1 and so on. This only doesn't happen if the difference is less or equal to 0.25.

AlwaysSame:

AlwaysSame's (AlwS) parameter calculates the investment very simply. The investment follows the equation:

$$i(\theta_{\text{AlwS}}) = \frac{\theta_{\text{AlwS}}}{10}$$

This means that after each incrementation of the parameter, the investment increases by 0.1. Parameter 0 is equivalent to AlwaysDefect and parameter 10 has the same behaviour as AlwaysCooperate. If the parameter is set to five, the strategy will always submit the investment 0.5 which is identical to Neutral.

Random-Continuous:

The other strategy in the group of the random is oppositely continuous, meaning it can additionally submit any number between 0 and 1. The name of this strategy is logically Random-Continuous (RndC). This strategy has a base being at 0.5. The parameter defines the shift being applied to the base, calculated by the equation:

$$i(\theta_{\text{RndC}}) = 0.5 + \epsilon \cdot s(\theta_{\text{RndC}})$$

$$s(\theta_{\text{RndC}}) = \frac{\theta_{\text{RndC}}}{20}$$

$$\Pr(\epsilon = 1) = \Pr(\epsilon = -1) = \frac{1}{2}$$

s means the shift. It is divided by 20 so the maximum shift, parameter being 10, can not exceed the range of 0 and 1. There is a 50% probability of subtracting or adding this shift, indicated by the symbol ϵ .

Random-Discrete:

Random-Discrete (RndD) is a strategy in the group of the random. It is also discrete, meaning it can only submit either 0 or 1. The parameter in this strategy determines the likelihood of submitting 0 and 1 respectively. The probability of submitting full defection is described in the following equation.

$$i(\theta_{\text{RndD}}) = \Pr(i = 1 \mid \theta_{\text{RndD}}) = \frac{\theta_{\text{RndD}}}{10}$$

or

$$i(\theta_{\text{RndD}}) = \Pr(i = 0 \mid \theta_{\text{RndD}}) = 1 - \frac{\theta_{\text{RndD}}}{10}$$

Parameter 0 is equivalent to AlwaysDefect since the probability of submitting 0 is 0%. And on the other hand, parameter 10 is equivalent to AlwaysCooperate since the probability of submitting 1 is 100%.

	$\theta = 0$	$\theta = 5$	$\theta = 10$
Random-Discrete	AlwaysDefect	-	AlwaysCooperate
Random-Continuous	Neutral	-	Random-Discrete ($\theta = 5$)
Always-Same	AlwaysDefect	Neutral	AlwaysCooperate
Adapt-Discrete	AlwaysCooperate	-	-
Adapt-Continuous	AlwaysCooperate	Tit-For-Tat	-

- Programming language:

The simulation, data generation and visualisation is completely written in Python (3). For the visualisation, as in generating the surface plots, I used a python library called Plotly. If anyone is interested in simulating the project on their machine, they can go on my GitHub page and follow the instructions there: <https://github.com/adho08/Prisoner-s-Dilemma>

4 Results

In the following two pages, you will see all the surfaces of both Random-Discrete and Random-Continuous. There are five surfaces that are generated from one ICPD. The x-axis, the one in be bottom-right, shows the parameter of the main strategy. The y-axis, the other one on the bottom, indicates the parameter of the opponent. The z-axis always shows the points. All the surfaces are not only marked by height but also by the colour. The context in which points are displayed depends on the surface. The first two rows are the most basic surface plots. They show the gains of the strategy itself and the opponent's. The highest point that can appear is at 30. And the lowest is at 0. The maximum can be calculated by regarding the pay-off equations. The maximum points are gained by exploitation. So, $P_A = y - c \cdot x + c$ becomes by substitution $P_A = 1 - 0.5 \cdot 0 + 0.5 \rightarrow P_A = 1.5$. And since the CPD is played 20 times, $P_A \cdot 20 = 30$ So the boundaries in which the points can vary is from 0 to 30.

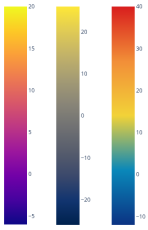
All the points, that are at height 20, have their correspondence in the other surface since in order to get 20 points, mutual cooperation is required. The same principle can be allied to the maximum and minimum points. Wherever a minimum point can be seen, the correspondence is the maximum point at the same x-y-coordinate. The colour of the points indicates the same as the height, the number of points. So, one can read this surface more exactly by regarding this colour scale.

A) The next two surfaces concern the advantage one strategy has over the other. This shows us how much more one strategy had over the other. For the surface of the main strategy, simply the opponent's surface is subtracted from it's surface and vice versa. The maximum and the minimum can be calculated by taking the difference of the maximum and the minimum of the first two surfaces. So, the points can vary between -30 and 30. ($30 - 0 = 30$, $0 - 30 = -30$) The two initial surfaces would intersect if they were put into one coordinate system. The intersection is at all the locations where the both strategies have gained an equal amount of points. At this location of the intersection, the surface plot, that demonstrates the advantage, would intersect with the zero-plane because the difference of two same number is always equal to zero.

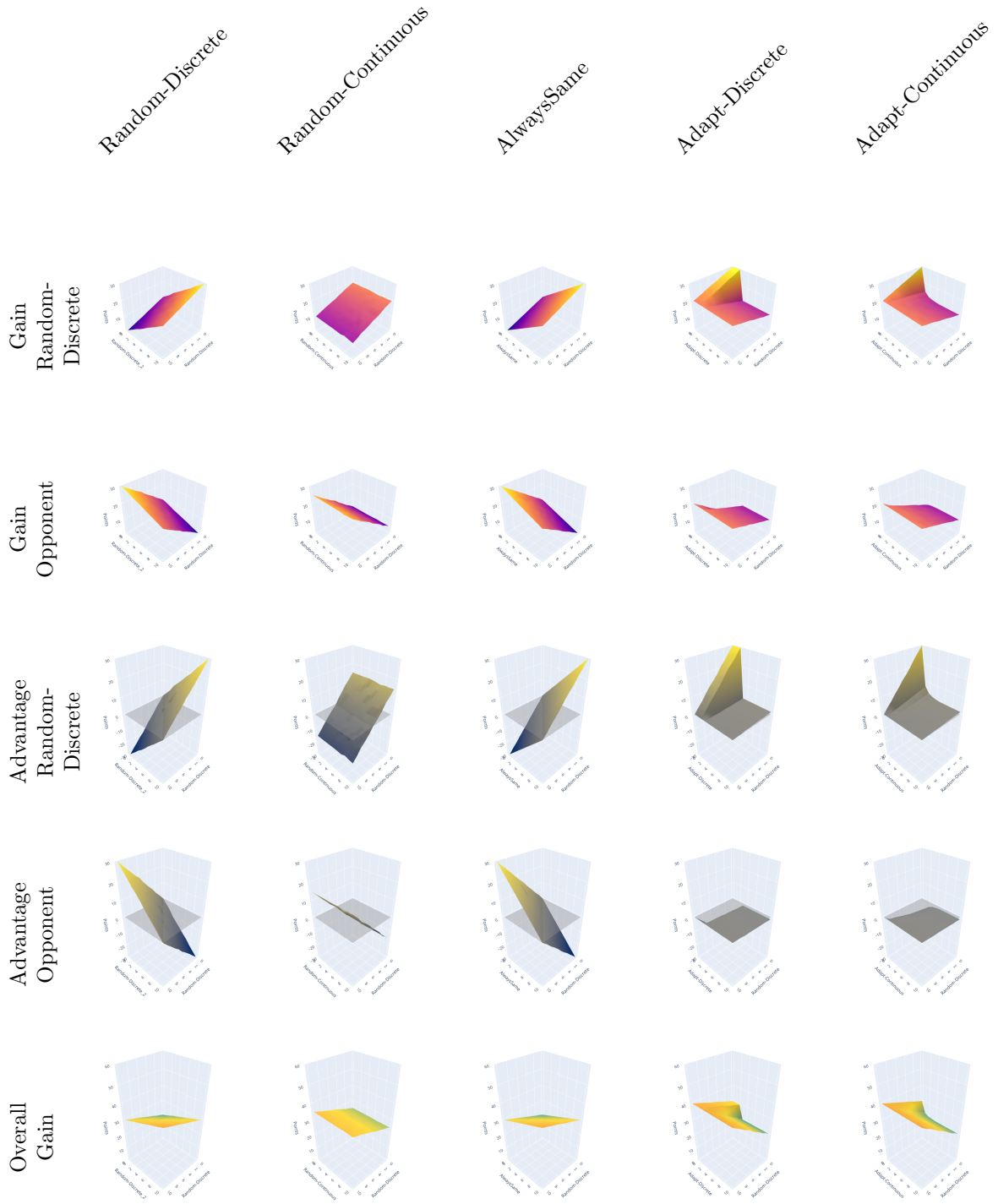
The two advantage surfaces are always mirrored by the zero-plane since every time you subtract $A - B$, it is the same as $(B - A) \cdot -1$.

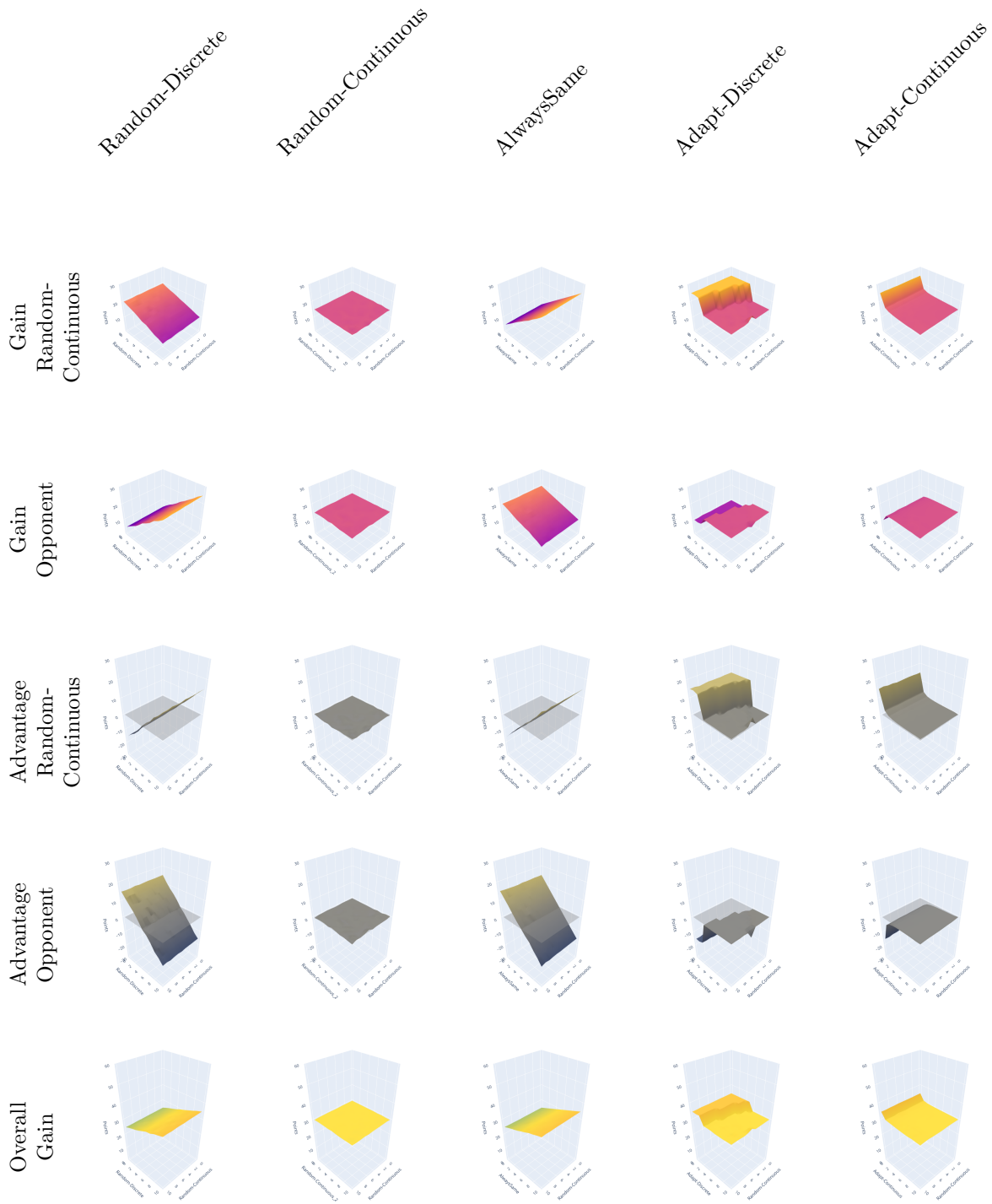
If this surface plot happens to be completely over the zero-plane, it means that the displays strategy has won every game they have played. Also here, there is a colour scale. This colour scale, however, differs from the first one. I chose to use different colour scales for visualisation purposes due to the fact that the advantage surfaces have a different z-axis range than the first two. (-30 to 30) B)

The last row of surfaces displays how much points have been gained by both. To generate these surfaces, one must add the surfaces which describe how much one has gained of the concerned strategies. Looking at the pay-off equations, a point can not be over the limit of 20. Both gain the most if both cooperate. Following the equations, one gets 1 point after every round. Since 20 rounds are played in total, the maximum is at 20. ($20 \text{ rounds} \cdot 1 \text{ point} = 20 \text{ points}$) I decided to also plot this surface to demonstrate the population welfare. This means that the higher the points, the most have gained both together. This plot has, as the advantage surfaces, a different z-axis range. So a new colour scale is introduced. C)



Random-Discrete





- Random-Discrete_2:

Gained points:

First, it has to be mentioned that Random-Discrete behaves like AlwaysSame. This is due to the fact that I have run this simulation 100 times to smooth out one-time abnormalities. This is notable by looking at the column where Random-Discrete plays against AlwaysSame. These two columns look arguably alike.

Since the two strategies behave exactly in the same way, these two surface plots are simply mirror-inverted by 45° on the x- and y-axis. The left and right corner are the extremes. Random-Discrete gets exploited when it has its parameter at 10, meaning it corresponds to AlwaysCooperate, and Random-Discrete_2 has its at 0, i.e. it is equivalent to AlwaysDefect. This is how one can get the maximum output, by exploiting. This can be seen in the surface of Random-Discrete_2. The behind corner is at the coordinate 0, 0. There, both defect constantly. This is the point where both get their relative lowest output, as can be seen later.

The nearest point is at the coordinate 10, 10. The parameter 10 indicates the same behaviour as AlwaysCooperate. So, both always cooperate and thus have not a so high output as when they exploit but fairly enough.

Advantages:

This time, the surface is mirrored at the cutting edge at $z = 0$. This is because both gain surfaces are mirrored. The absolute peaks of both derive from having the maximum output. Since one can get the maximum output only if the other gets exploited, the advantage surface result in the highest peak possible. When the two parameter's are identical, however, they are not any better than the other. This makes sense since they behave the same when they have the same parameter and can thus not exploit each other.

Overall:

Here, it is clearly visible that the high point on the surface is when both cooperate. This illustrates the dilemma pretty well. It is better for both if they both cooperate, all the time. As already mentioned, the most behind point, being the lowest, results from the fact that both defected constantly. So, for the welfare of the population, it is best if both cooperate every round.

- Random-Continuous

Gained points:

This surfaces ranges from 20 to 10 continuously. It seems as the surface is not influenced by the parameter of Random-Continuous. This is logical and will be looked at the discussion.

The second surface ranges from about 5 to 25. Also here the surface is only influenced by the parameter of Random-Discrete.

Advantages:

Both surfaces, when put together in one plot, would intersect exactly in the middle. Since they range from different limits but both upper and lower limit add up to 30, they intersect in the middle where the parameter of Random-Discrete is 5. That is why the surface of advantage intersects the zero-plane at the same line.

Overall:

Naturally, this surface is only influenced by the parameter of Random-Discrete and not Random-Continuous. It ranges continuously from 25 to 35.

- AlwaysSame

This column of surfaces looks identical to the one against Random-Discrete_2. In the discussion further investigation will be made. Since I have explained the properties of these surfaces already, I will not explain them again.

- Adapt-Discrete

Gained points:

In the surface of Random-Discrete against Adapt-Discrete, there is a stripe at the parameter of Adapt-Discrete being 0 to 2. The stripe ranges from absolute exploitation at (0, 0) to mutual

cooperation at parameter of Random-Discrete being 10. Abruptly, this stripe does not follow its pattern in the x-direction any more. Rather a new slope is visible. This slope has its lowest points at 10. But, both slopes range from where they begin at the y-z-plane to 20 at the other y-z-plane. Remember that 20 is the result of mutual cooperation. As a result of that, there are also 20 points in the corresponding surface plot at the same x-y-coordinates. This is indeed the case. Also the maximum points of the first plot indicate that the same points on the other surface must be at height 0. The broader slope is also going downwards. But this slope is not as steep as the one on the other surface and thus has lower points at the edge where the parameter of Random-Discrete is 0. When Random-Discrete has its parameter at 10, which means it behaves like AlwaysCooperate, mutual cooperation will take place.

Advantages:

Since the slopes look pretty similar, there are practically no advantages in this area. At the coordinates of the stripe, however, is a great advantage which only Random-Continuous has over Adapt-Discrete. The advantage surface of Adapt-Discrete is not visible to see relevant part. The plane covers the inverted stripe of the advantage surface of Random-Discrete. The rest of the surface is very near to the zero-plane which means it was as good as Random-Continuous in this area. Overall:

Of course, the maximum points are at the locations where mutual cooperation is held place (40). This stripe is, this time, not as steep as the broader slope. The slope ends at the y-z-plane at height 20.

- Adapt-Continuous

Gained points:

Intuitively, the discrete stripe becomes a continuous curve. The edge where points are at height 20, stays as it was in the game against Adapt-Discrete. The broad slope is also similar to that surface, its lowest points being at 10. The surface of Adapt-Continuous is in a same way similar to the one of Adapt-Discrete. The edges are curved out. But, the slope is, again, a bit steeper than the one of Random-Discrete.

Advantages:

The same applies at the advantage surfaces. They simply are curvy instead of edgy. The advantage surface of Adapt-Continuous is in a difficult angle to see. The origin of the coordinate-system is the lowest point of this surface. I concluded this fact by knowing that the highest point of the other advantage surface has its correspondence at height 0.

Overall:

The maximum points are aligned in a line. More specific, an edge of the surface at height 40. The least points are gained at the edge more behind at the y-z-plane. This edge is rather a curve than a straight line. This curve bows from exploitation at coordinates being at (0, 0) to 20, the absolute minimum of this surface.

- Random-Discrete:

This surface has already been explained. These are the same surfaces as when Random-Discrete played against Random-Continuous in the first figure.

- Random-Continuous_2:

Gained points:

As already mentioned, Random-Continuous behaves like Neutral by having played several times. So, Random-Continuous will submit 0.5 always. This results to having a completely flat surface. We can calculate at which height the surface is. Plugging in 0.5 for both x and y in the pay-off equations, we get $P_A = 0.5 - c \cdot 0.5 + c$ where c is, as always, equal to 0.5. The pay-off of strategy A is $P_A = 0.75$ and having played the ICPD 20 rounds, it is equal to $P_A \cdot 20 = 15$. Both strategies A and B get the same pay-off since they submitted the same investment. $P_B = P_A \rightarrow P_B \cdot 20 = 15$. So, both surfaces are at the same height at every point.

Advantages:

Subtracting two flat surfaces at the same height at every coordinate, gives you a also flat surface which corresponds to the zero-plane. No strategy was better than the other at any point.

Overall:

Adding these even two surfaces which are both at height 15, gives you a surface at height $15 + 15 = 30$.

- AlwaysSame:

These surfaces are identical to the one's of the Random-Continuous vs. Random-Discrete interaction. I will not describe them as they already have been.

- Adapt-Discrete:

Gained points:

There are two main levels on this surface. The upper level in the Random-Continuous surface is at height 25. The lower one is at height 10. There is a little peak at the coordinate (5, 10). The two levels are joined abruptly at a horizontal curve.

The surface of Adapt-Discrete is in some way inverted, meaning the upper level in the other surface corresponds to the lower level in this surface plot and vice versa. This lower level is at height 10. The upper level of this surface and the lower level of the other surface are at about the same height as we will see later (height 15). Also, the peak is now a pointing downwards.

Advantages:

Naturally, the upper level of the first surface indicated that it semi-exploited Adapt-Discrete. This is now visible in this surface due to the fact that this upper level is very high. It was at every time on this level 15 points better than the opponent. But, the lower level is nearly at height 0. This is what we will see also in the next plot and that is what I meant when I said that the lower level of Random-Continuous and the upper level of Adapt-Discrete are about on the same height. This level is, however, over the zero-plane, meaning that it was better in every single game they have played.

Here, we see that this Adapt-Discrete was semi-exploited. The upper surface is a bit lower than the zero-plane. This means that Adapt-Discrete was The peak of course is also inverted.

Overall:

According to this surface, the upper level of Random-Continuous added to the lower level of Adapt-Discrete higher than adding the two other levels with each other. The peak indicates that also the peak of the surface of Random-Continuous is higher than the peak of the surface of Adapt-Discrete is low. Only with this condition can this peak point upwards.

- Adapt-Continuous:

Gained points:

Random-Continuous's points range in a form of a wave or a curve in the y-direction. This derives from the fact that, as already mentioned, Random-Continuous behaves like AlwaysSame with parameter 5. So, in average it submits 0.5 all the time. The highest points are along the edge of the x-z-plane which are at height 25. Following in y-direction, the lines approximates to the plane at height 15 and at a certain parameter of Adapt-Continuous, it seems to have reached this plane. From that on, it follows the plane until it reaches the last parameter of Adapt-Continuous.

This surface looks exactly the same with the notable difference that the curve is now pointed downwards. The curve begins at height 10 and approximates a plane at height a bit less than 15. The plane, this curve approximates, is in a bit darker tone of colour than the other plane. This means that this approximated plane is a bit lower than the other.

Advantage:

This surface looks exactly like the first surface in this column, it is just a bit more extreme. The curve is starting from 15 and diverges in the plane at height nearly 0.

This curved surface ranges, naturally, from -15 to nearly 0. One remarkable feature that these surfaces have is, as the one's a column at the left, they do not intersect the zero-plane. This means

that Random-Continuous has gained more points in every game. Adapt-Continuous is objectively worse since it lost every game.

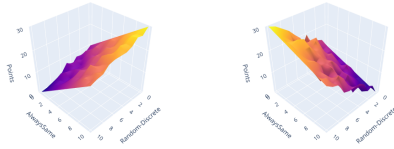
Overall:

This surface shows us that the peak edge of the semi-exploitations added up with the exploited strategy's lowest edge is more than both submitting a fairly similar amount of investment. Remarkably, the whole surface is pretty high. The maximum is not reached but every point is fairly high.

5 Discussion

- Random vs Rigid:

I have spoken of the point that Random-Discrete, as a mixed strategy, behaves like AlwaysSame when taking the average of having played the game several times. In the simulation, all games were played 100 times and taken the average to plot the surfaces. But, of course, when only playing once, the surfaces would look different.



These surfaces show the gained points of Random-Discrete and AlwaysSame when playing the ICPD only once. A rougher surface is visible. This is due to the fact that Random-Discrete uses probabilities after all. Whereas AlwaysSame submits always a determined investment.

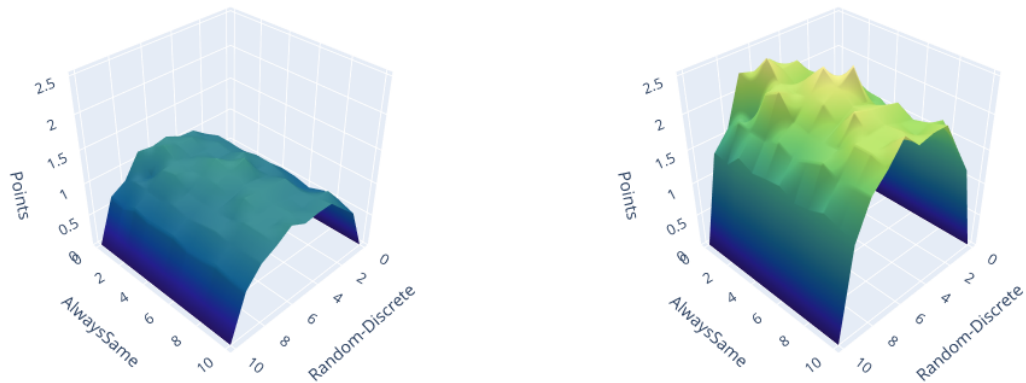
The surface of Random-Discrete is a bit even. This fact can be explained by examining the pay-off equations: $P_A = y - c \cdot x + c$ and $P_B = x - c \cdot y + c$

x being the previous investment of Random-Discrete and y is equal to the previous investment of AlwaysSame. x is relatively random, depending on Random-Discrete's parameter and y is in one game always constant. Subtracting only a fraction of the random investment pays off a more constant amount of points.

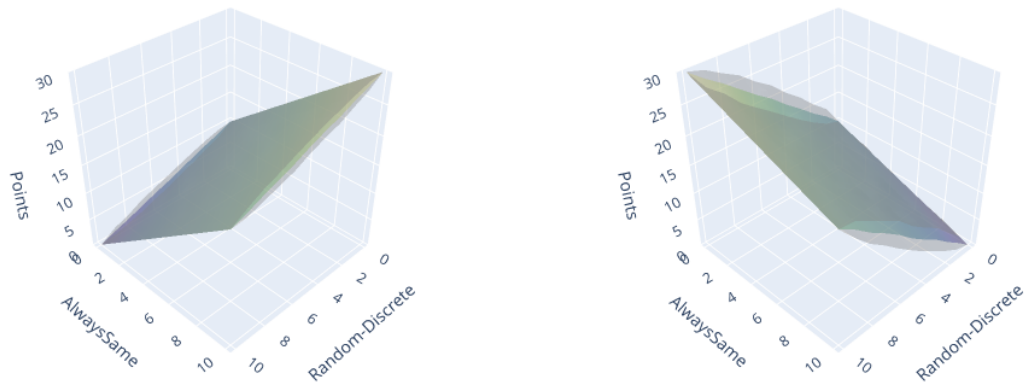
To calculate how much Random-Discrete derived from its constant-behaved opponent, we can use the standard deviation for every game. This means we can display a new surface plot which will describe how much off the investments were comparing to the average. This formula is being used, the standard deviation.

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

We get the following surface:



To conclude, choosing Random-Discrete, especially parameter from 2 to 8, as one's own behaviour means a high risk of losing. But, this risk of deviating from the average is at every point twice as much with AlwaysSame than Random-Discrete. This means that when playing like Random-Discrete, one risks getting \pm the standard deviation to the average. But, one is applying twice this risk to the opponent which is in this case AlwaysSame. This can be seen in these two following surface plots.



(Why chosen AlwaysSame?: Because AlwS and Rnd-D seem alike, but aren't)

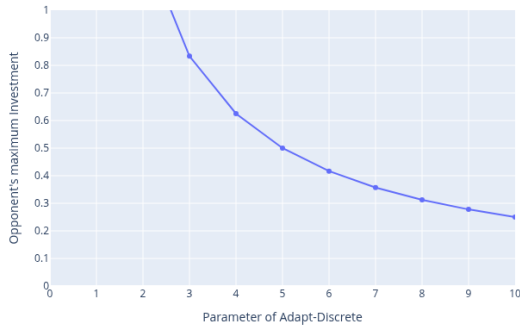
(3D object in which the prob. is 66% of being in there in one match.)

(Why are Standard Deviation surfaces same but different scales?)

- RandomD vs AdaptD:

When you look at the surface, this stripe is very remarkable. We will look at why this stripe exists and why is only exists from AdaptD's parameter 0 to 2.

As Adapt-Discrete is explained, its investment is rounded to the nearest integer. So, in order to get from investment 0 to 1 or vice versa, its investment has to get over the threshold of 0.5. To demonstrate this point, the following graph is used. It shows what the opponent's investment has to be at least or at most to surpass this threshold.



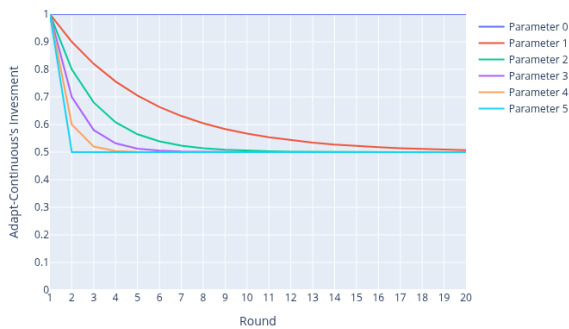
The derivation to calculate this function is the following can be found in the Appendix. It can be seen that if Adapt-Discrete has the parameter from 0 to 2, the opponent's investment must be over 1 which is not possible. This explains why Adapt-Discrete with its parameter set to either 0, 1 or 2 behaves exactly like AlwaysCooperate since it cannot surpass the threshold of 0.5.

Coming to the next attribute of this surface; the evenness of the rest of the surface. The other part of the surface is first, not influenced by the parameter of Adapt-Discrete and second, has a little inclination. This question can be answered with the fact that Random-Discrete behaves, as mentioned many times, like AlwaysSame. The two slopes look nearly identical because Adapt-Discrete will also behave rigidly against a rigid strategy. A little detail is that the slope of Adapt-Discrete has a very little larger inclination. This means this slope is a little bit steeper. This mysterious can be answered by the fact that Adapt-Discrete always starts off the game with full cooperation. So, the first time, Adapt-Discrete gets a little bit more exploited the nearer the parameter of Random-Discrete gets to 0. Because at parameter 0, Random-Discrete is like AlwaysDefect and gets thus fully exploited. Random-Discrete's parameter 10, in contrary, it like full cooperation so it get not exploited. This means that it gets more exploited the nearer Random-Discrete's parameter gets to 0.

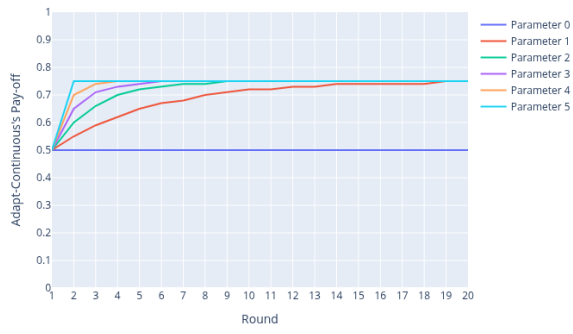
- RandomD vs AdaptC:

A same idea can be used to explain why this surface has a similar appearance like the one before. The following line chart demonstrates the investments of Adapt-Continuous in one game in the ICPD. I assumed that Random-Discrete behaves exactly the same as AlwaysSame. So, Adapt-Continuous pursuits to get to the same level of the parameter of Random-Discrete.

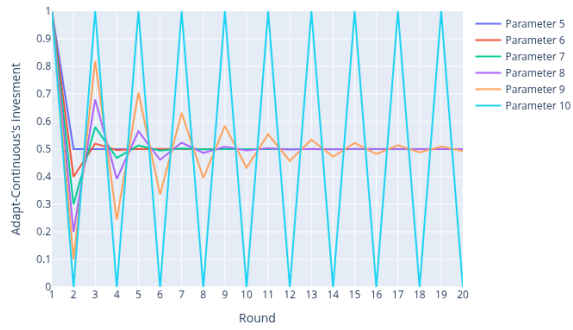
Random-Discrete is, in this chart, at parameter equal to 5.



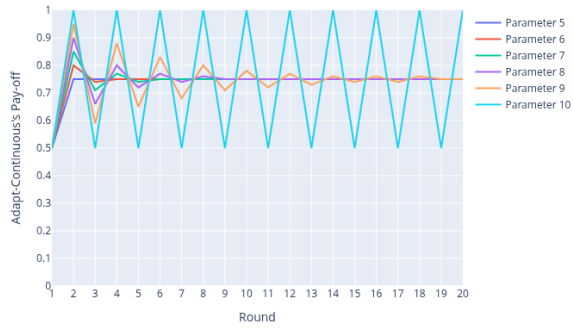
You can see that the higher Adapt-Continuous's parameters are, the faster they can adapt. This means that an Adapt-Continuous with a low parameter gets more exploited in the first few rounds than a Adapt-Continuous with a high parameter.



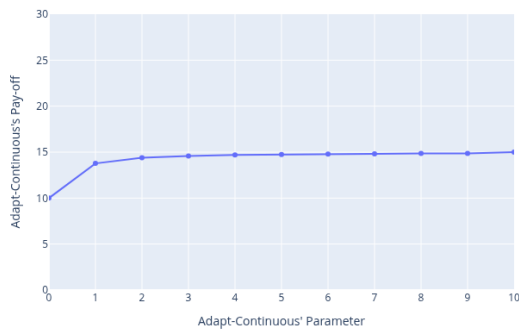
The question arises what the behaviour looks like if Adapt-Continuous has its parameter over 5. The next plot shows exactly this.



The pay-off's of these investments, however, cancel out.



The peaks and the lows at parameter from 5 to 10 converge into an average. These averages can be seen in the next plot.



To summarise, from parameter 0 to 5, the pay-offs get better since they can adapt better. But, from parameter 5 onwards, the pay-offs stay exactly the same since their pay-offs converge to an average.

- RandomC vs AdaptC:

6 Conclusions and Outlook

- Summary and Quintessence
 - which strategy 1 better than which strategy 2
 - which parameter to be used
- Application in the Real World
 - strategies translated to real persons
 - giving examples (Politics, economics to simple human beings)
 - prospect to the future (which ones will win?)

7 Self Reflection

- Challenges
 - estimation of time consumption
 - transforming ideas during process
 - dedication
- Learnings
 - reading professional papers
 - writing English
 - experimental workflow (usage of neovim, tmux and terminal)

8 Annex