Dimensional Game Theory By Nnamdin

Okay guys, welcome to dimensional game theory from chess the Unicode. Okay, guys, this is important to me. I want you guys to take with almost dignity.

It's not to diminish game theory that it is. It's to expand the horizon of game theory in the real world. You know, dimensional game theory is the real world's game theory, you know.

And it's based on dimensions, you know. This project is to cover all you need, just to get started with dimensional game theory. But it's not to dismiss game theory, and it's also an extension of the game theory framework already here.

You know, it addresses real world problems, you know, real world scenarios, you know. That's the problem that dimensions have, the power of the dimension. Now, the dimension is something you define, you know, to address this problem, you know.

I'm going to give the limits explanation, and I'm going to go through this. Okay, so I got my, here, okay, basically here, I got this. So, my pen here, red, orange, so my ink here, you know, my pointer.

So, dimension, the dimension is, in naming terms, dimension is a problem you define for what you want to enforce in the game, you know. It's like, you know, red team, blue team, what's the definition, what's the particular definition, you know. In chess, you know, this attack dimension, defence dimension, and offence definition.

Now, these dimensions represent problems, you know. I want to attack, I'm going to give the champions to attack. I want to set up an offence with a pawn.

I want to give defence, more defence, defend. What is the strategy? You know, there's got, you got a lot of strategies like the knight, I can't remember the name, the lover. You got the rook strategies, you know, like other ones you can use.

Those are fragmented, you know. Because, you know, the attack dimension overlaps with the defence dimension, you know. You don't have a perfect attack, you know.

You have something that's attack, but it's doing defence at the same time, you know. Or sometimes it's defence, or offence, doing attack, you know, vice versa, across all dimensions. One dimension with a game theory, oh, that's what I ordered you to call it.

I'll be sure to call it. Okay, yes. What dimension game theory really gives you? You know, in a solid framework for isolating those algorithms in the attack, defence, or offence, you know.

I'm creating new dimensions that resolve the direct exactly grow up. So, you can perform each dimension and evolve the dimension in these threads when you use these animations, you know. That is the first foundation I want to give you.

Okay, let's continue. Content, the big idea, the big picture, you know. No, the big picture of the mission game theory is never to the mission, because it gives you, you know.

The big picture is what some, you know, in numerous terms, it gives you the dynamic strategies, you know, dynamic. The real world is dynamic, you know, model that, you know. It's model real world systems, you know, real world games, you know, as dimensions.

These are called dynamic algorithms, you know. Or I'll call them a lot of original algorithms, you know. Because they evolve with human cognition, you know, you know, and consciousness, you know, new human position, you know.

I call them evolution strategies, you know. These strategies evolve based on the information, you know, who will learn the enemy loop or what's something, you know. It's about ensuring that, you know, the system actually has a goal, you know.

And that goal is based on dynamic adaptation in real world context, you know. It allows for you to be authentic in expression of game theory. Of game theory without being bogged down with static linear systems, you know, that grow out of linear bound complexities.

Let's continue. Now, let's check the structure, you know. Go look at the big picture.

No, this is going to use dead projection against UK security. Advancement, algorithm call and the robot applications. Okay, guys, let's go.

System integration. And take away the context. Okay, part one, the big idea.

Dimension of game theory, the size of perfect strategy. From chest algorithms to hacking Unicode, a framework for competition, you know. Okay, basically.

Basically, this one's saying to you. The chest algorithm and Unicode hacking are the same, you know. This competition, red-boom to business matter and the rule of thumb, you know, you're going to have enemies.

You're going to have people want to infer to your systems. You know, you can model them. You can model a parallel algorithm in one dimension.

But you're going to grow. You're going to grow in a competitive balance, you know, dimensions. You're going to neglect other dimensions.

If you think the algorithm is perfect, you know, it only does one dimension, you know, not such an attack. But it doesn't do the general defence. But it's offensive ways to explain the difference.

You know, this one, dimension of game theory, really has to address the balance of entropy in the system. In the real world, you know, the balance of the system. Okay.

The limits of cognitive history, okay, static and rigid. Realised on fixed payoff matrices and equilibrium concepts that struggle with dynamic real-world complexity. You know, this is a big thing.

Back there for the classical development. Classical degree is too static. It's too rigid.

Every input has an output, you know. But that input never evolves. You know, my business adoption evolution.

The fixed payoff is like, you know, basically, it's like, you know, if I have someone knowing what I'm doing in the real world, you know, I know what they're going to do, too. So if it's a tie, if it comes from the equilibrium, which is like a tie, you know, it's a tie, you know, I know. Then I start looking at my, you know, 60-50.

Then I risk it or lose it all, you know. Then my head, I'm thinking psychologically, I'm trying to risk everything, you know, because this guy is onto me. And the other guy is thinking, yes, I'm going to risk everything, you know.

And then it comes to tie. Or in the other case, it comes to the worst one, you know. They don't risk anything.

And they both lose, you know. You know, because there's no clear distinction between who's going to lose and not win. Because, you know, the game is an equation of a tie or equilibrium anyway, you know.

Because, you know, that's the model of the chess and everything. The thing is, it fails to come, new parts are suitable. Basically, there's new algorithms, new ways of doing stuff, you know.

And there's many ways we can come up with scenarios to problem using other theories. You know, all the scenarios, you know. You know, they can directly do it.

Or they can use adapted algorithms to create new solutions to the problem. You know, rule complexity is a big thing. But dimension and game theory addresses that using dimensional models of that strategy.

We need a rule of three and practise. And that seems a grounds to presume. It's too rigid.

It's too static, you know. There's nothing, you know, you can predict input from the output, you know. It's too mathematical.

It's too slow. It's not slow. It's like, it's slow, but it's too box.

It allows you to, the structure is too rigid, you know. There's no flexibility even for the user who wants to stay in that framework. Because the classical game theory does not adapt for their context.

Because the inputs never are never more vectors. They're just dimensions, no dimensions. Just single layer dimensions.

No, that's not a classical. Just single layer. Just killer dimensions.

Only what inputs, you know. Strategy dimensions are the axis of play. Decomposing any gun between dependence non-overlapping axis, each dimension can be optimised separately while contributing global carrying his strategy.

You know, basically, it's like a sound mixer. You don't have to, you don't, you don't have to, you don't just have the volume. You have the bass, treble, and mid-range, you know.

This I'm saying, you know. You have, basically, it's like a sound mixer. This dimension represents a purpose.

Like, you know, a master volume. It's just for the volume. But, you know, you can tell by the bass, you know.

You can tell by treble. You can tell by the mid-range, you know. All the songs that make the sound more in tune with the environment.

You know. The thing about dimension game creating. Once you analytically express those things as individuals, you know.

As a core, as core individuals. Like, you know, there's no, there's no thing about it, you know. The game is here, strategic, you know.

It allows for flexibility. Because the dimensions are isolated. They're independent.

They're independent. Don't develop over the axis, you know. The dimension is like, you know.

Attack, attack algorithm is not a defence algorithm. Defence algorithm is not a defence algorithm. Defence algorithm is not a, it's not a, it's not a, it's not a formation algorithm in chess, you know.

You know, accessory, good dimension algorithms. How to organise the best structure. Get the perfect game.

And not gonna overlap them. Okay, the perfect game theory. In the zero sum game.

With perfect information. If both plays are in place of more strategies. You know, dimension.

The game will result in a limited time. You know. That's the truth, you know.

You know. That's the truth, guys. You know.

Guys, you know. This is no joke. A non-lie outcome to prove a strategy imbalance.

This is a signal we can exploit. A corollary. What's the corollary? A non-tie outcome is proof.

Of structuring imbalance. This is a signal we can exploit. You know, that's a corollary.

You know. It's not an axiom. It's like, you know.

It's not. It's an axiom, yes. But it's a starting point for addressing this perfect game problem.

You know. Because, you know. It's like.

I mean, chess is a perfect game. You know. And he plays for the chess.

And I play for the YouTube. The game is a tie. It's a tie.

You know. You know. And the game always be a deterministic tie.

DFA tie. No. Not an NFA tie.

On a system. Nothing there. You know.

I don't know NFA yet. But it's a DFA tie. It's just like the system.

Understand you play perfectly. I play for the game. You know.

It doesn't matter what the game was. You know. And that's why the perfect game theory matters.

No. The theorem is true. It's a better term.

It's true. You know. It applies to all games.

In the end dimensions. You know. Where we do.

But the corollary is. If we have a strategic imbalance. In any other dimension.

I play more than attack. He plays less. Basically.

If I play a perfect defence. And then he keeps. He plays a wide range.

No. Not so perfect defence. You know.

I have. I can couple this with my attacks. The balance.

In the defence. Structure of his. Or vice versa.

You know. It doesn't matter. This one is saying to you.

DAG. So basically. Directed exactly.

A graph of this chart. You know. There's no overlap.

Now. Directed aside graph. It's a structural backbone.

You shoot the winning competition. And independent applause logically. Preserving.

Other. Overall strategic clearance. You know.

Basically. The DAG framework. It's like you know.

And. Can you trust. Do the cost function.

Hamiltonian. Or Larian. Cost function DAG.

Based on real world estimate. But you know. To get to the goal.

You know. You know. It's like.

It's like something in the real world. It's like. Trying to.

Yes. It's actually. Again it's objective.

Done. You know. In time.

But the time. It's not the goal. It's time.

It's not the goal. It's like a traffic. You know.

But you have to. Check the. Check the road.

B C. For example. And then go down. Say it's like G. Because I can go like.

Basically I'm going to A to D. Basically this car. Can't go there. A to D. But it has to go all the way.

You know. So it has to. Excuse me.

The cost of this road. You know. And that road in the real world.

Preserves the logic. A to D. It's going to. It goes A to D. You know.

But you know. But this is an imbalance. It has to go.

You can go A. You can go A here. Just depends. Whatever is shorter for it.

What is shorter? What is missing? The traffic cost. The jam. What it knows about the state.

You know. Can't spot a dark access ramp. There's no overlap.

You know. This should never open up in a dark structure. You know.

Because you know. It's not going from a dark area. Unless the system.

Doesn't allow. Doesn't allow for the path to go. You know.

The time that the edges. Can represent. The cost of the.

It can basically make a cost to get A to B. You know. A to D. But you know. If the goal.

If there's any determinants. Which is worst case measurement. Of the dark structure.

For the dimension of the game. So I'm putting new strategies. You go from E to E. Or whatever is available in that dimension.

You know. As an optimal log n space. You know.

The computer is log n. It's half time. Space. Because the space is like the storage.

Space of the storage. You know. It doesn't have to think.

It just stores the algorithm. It's log n. You know. It's actually spaced.

It's like you know. The front equation. Of the log n. It's half the time.

It's never linear. So you can count the problem. Any problem in half.

I know some claiming here. Because. The dimension of game three.

Is independent. Of the. Of this.

Is independent. But logically. We are the directors.

Algorithm resolution. You know. You can just see the question.

Attack defence. And offence algorithm. Or for me.

Get a new formation system. No. No.

Sorry guys. Sorry guys. I don't know about.

Sorry guys. Sorry guys. Sorry guys.

Sorry guys. I got interrupted. The turn is playing.

Okay. So guys. So basically.

I didn't want you guys to hear that. Sorry guys. I didn't want you guys to hear that.

So Z team. They don't understand about formation. They don't even need formation dimension.

You know. That can reject chest formation. You know.

Or something. Or car. On the formation breaks.

You know. The formation break down. They have to get someone else.

There's a new ES system. A new car. To drain that formation.

You know. And that formation has to be aligned with objective. You know.

That formation is not the chest. All of the car is doing. You know.

Your chest are calling them. Getting to the objective. You know.

That has to go. You know. To ensure there's no delay.

You know. That's what I'm saying about the cost function. The dagger of the graph.

You know. Because you know. The piece of the car.

It's not missing. But someone is new. Is scalable.

You know. It's scalable. You can scale it.

Isomorphically. With peer to peer. Bar string topology.

Which are four thread models. Of topology structure. You know.

For real world context. You know. Dimension of green.

To represent to me. The real world applications. Of computation.

In the system. That resolve autonomously without any humans. You know.

Because you know. If I order an Uber. An Uber is a robot.

You know. His robot. And she gets the phone.

You know. If I have a premium member. You know.

But you know. If I'm not a premium member. And I still need an Uber.

You know. I still need. I need to get an autonomous Uber.

And get that Uber. Stop. You know.

Because all the parameters. Stop. They are an additional problem.

You know. Because I am entitled to that service. You know.

You know. I can get that car. And I can go.

You know. That's what I'm talking about. The cost function.

I have the direct. A sacrificial position. For the real world cost function.

Metric. To ensure that the 5.4. Coherence. Over function.

Composition. Which is the cost. Moving the real world.

Over terrains. And places to go. NFA function model.

Place a model. NFA. Okay.

Place a model. NFA function model. Actors.

That means they have way to act on a decision. You know. Whether they see.

Exploit or something. In the real world. They are actors.

You know. That means they can act. They not just.

Not so much. You know. They can be.

You know. They can be. But the.

The actor model. It's like you know. You don't have an insight.

In something. I can still win it. But you know.

It's like a plain dimension. Like I say it's an NFA resolution. You know.

But the thing about NFA is like. It's there. You know.

It's not going to sound like something. But you know. Map the function to state action.

You know. The allows algebraic composition. A model.

Basically the actor model. Is more of you. Acting on your own behalf.

On your own code. You know. Autonomously.

Your autonomous. You know NFA. You know.

It reminds me of delay. Deny and defer. The defer is one dimension.

To deny. Deny to right housing. So I can get denied housing.

And deferral is go somewhere else. And deny is go. The defence occur.

We didn't do anything. It's not affords. You didn't sign up.

You know. That's algorithm. Now.

The NFA resolution. Is key to this. Because you know.

That's to end is to it. You know. If I ever.

I said the. It's actually go somewhere else. Defer me.

I will lose my housing. And if I really defend my business. And I say no to them.

You can't. So it's my fault. I'll lose my housing.

And I don't think delay is. Blame me. Waste time.

Go somewhere else. Deny. Deny.

You know. It's not affords. You know.

Deny. You know. This is what algorithms you know.

Delay. Stop waste time. Deny.

Say no. Say no. It's not a job.

It's not a society. Defer. Go somewhere else.

You know. I'm saying. See.

The NFA function can resolve this DAG function. You know. Because the function is a mapping of function.

You know. So if I want to delay some. Defer someone.

Or deny someone. Which is illegal treatment. I have to say.

Okay. I have to make it. Then.

Understand DAGs original. Making. But you know.

You must be sound coherent. To be coherent. And to work.

You know. That's to make sense. And that's why the function.

Mapping a functional model coming. Which can act on your own behalf. Since that's our structure.

You know. Basically. The structure of the system.

You know. It's like the. It's like how the system.

Understands. It's. You know.

The structure. Over the syntax. You know.

The core results. But the structure. Of units.

You know. The synthesis. Just rules.

You know. But you know. If the structure of that system.

Is. Based on. The syntax is not a structure.

Because. The. The structure.

Is like the architecture built on. But if that syntax. Doesn't.

Allow the architecture. There's no. There's no meaning.

Can I get system to do any extra. You know. So you build first structure.

You're building the architecture. The infrastructure you're building. Before you.

Jump into the code. You know. The perfect game algorithm.

Emerges. When structure reflects through understanding. You know guys.

Structure. Structure is more important than syntax. You know.

You know. No matter how you think about it. The structure of this structure.

You know. Or the model. Or something you're building.

You know. How you model the real world. Instance.

You know. How much clarity you have in that model. You know.

As a foundation. You know. Of that system.

It's over the structure. Of the syntax. Over the analytical framework.

You know. It's more beneficial. Because it's like.

It's like security. It's like my housing thing. Security by design.

You know. It can design. The perfect window.

The topology. And the compound. You know.

And even the modes. No one will go to your modes. You know.

Because you know. This is the shock. And then you can get your modes.

You know. The structure of your modes. In the compound.

Allows you to get in. And get out of your home. You know.

But the structure of the topology system. Is that. You know.

You know how to get to your house. You know. As a security measure.

But you can't get in. Because the structure reflects. You towards.

A purpose. A security. Of like you know.

The perfect algorithm. You know. Because you know.

It's like designing the houses. Your house. To be.

To be naturally hostile. To anybody's intruders. You know.

Without. Because you know. It's not trying to get your house.

It's like a helicopter. Or something like that. To get to your properties.

But it's not feasible. You know what I'm saying. It's not easy to do.

It's feasible. It's not easy to do. But.

But. Part ends. There's cross.

There's cross section. See. Case study.

The perfect chess engine. Okay. Now.

Dimensional game theory. Decomposings. Chasing to four.

Overlapping the short contents. You know. Each opportune of four.

Rejectorability. Semantic resolution. Detect dimension.

Capturing threats. King pressure. Protect the morphics.

This is like an attacker. How to get capture. You know.

The defence. Protecting kings, pawns, structures. Skip routes.

You know. And then offence. Spacing.

Piece activity. Initiating. This topics is more analytical.

But you know. You can use the offence to your advantage. You know.

But you know. All of this is analytical. You just have to solve the problem.

Conditional harmony. Coherence. And mobile pattern.

Mobility pattern. You know. That information.

Like you know. Ensuring that you know, you risk a pawn. You know.

Something. You make it. You make it.

You risk pawns. Not the fool. A fool work or something.

You know. Or something that can. The bishop.

Or something like that. Like a queen. Or something that the knights.

Something that can be used in another context. You know. You can call it like some sort of defensive field, you know, it doesn't know what you prioritise per se, you know, you know, but you can practise or attack what you want.

Because we are lacking any of the other things, you know, there's an imbalance, there's a drag, there's actually a second-graph imbalance. The system will fail, ultimately, because you neglect all the other ones, you know. I don't think strategically, I mean, I don't know the dimension if you want, you know.

The dimension is like the only isolated dimension, they never overlap. They never overlap. You can quickly imagine whether you're modelling eyes, but you never overlap.

Attack and defence, the core tactics. Now, attacking an algorithm, defence, let me see this algorithm. F A, function A, function A, function D. Now, this is F, the defence, function for defence.

The text immediately captures, you know. This is like a pose, you know, analyse speed threats, compute skin pressure, identify tactics motif, you know. The motif is like the payoff of, like, what the enemy is going to do, but I'm going to exploit that with an attack.

It can be attack, attack the territory of here. But this, like, algorithm can be here, here, because these are like these functions, you know. You can add inputs, which is a pose, you know, but they organise, they analyse speed projections, evaluate the key safety, assess point structure, you know.

These are dimensions, you know, different definitions. The four, this change, these are decisions, you know. This is a set model, the decision set, you know.

They're not even set, you know. Can you unify one dimension? Never use your hand, you know, or keep the dimensions isomorphic to themselves, because this is a problem you solve in one dimension, another. You don't need to, basically, you don't need to fragment the system.

There's no fragmentation whatsoever, you know. The offence versus formation. Now, offence is about long-term pressure, space control, peace activity.

It's about slowing the enemy down, you know, slowing the defence down, or slowing whatever piece. Offence is more strategic, because it's about the slowness of the system, trying to control this design of what's happening on the board, but, you know, it's invisible. It's like that, it's like, you know, before you know it, you use a piece that you don't know, you know, but, you know, it's about picking the right fights, you know, and ensuring long-term sustainability of the system.

You know, for information, it's the harmony. It's like, you know, collision, it's like, you know, it's like a, it's like a, it's like a, it's like a porn. But that's not right, queen.

This formation is not good. That means the formation is not good. Because, you know, this is slightly imbalanced.

I don't, I couldn't keep my formation. That means the travel formation. Because the solution didn't allow me to do my strategy, you know, you know, or do what I do with my formation.

Because the information is intact, you know. The pose can be lost, or I can throw the pawns, or the pawns, or the little pieces, and keep all the big pieces in the game, you know, as a perfect formation. You know, because then there's a perfect formation, I resolve that.

And then the offence is like the space control, making the enemy weak, weak, weak, you know, and making an example of the pieces they, the moves. And this is putting them psychologically, you know, you know. Red blue page, basically, this is like a move.

No, no, no, no. This is a change in PR, probability of photo blue page is 0.6. Because, you know, this is an imbalanced vector. You know, this one says, you know, 7 minus, 2.7 minus 1 is 0.6. You know, the largest probability of attack dimension, signals the collections of the advantage in red, you know.

Because, you know, these dimensions represent attack, defence, offence, and information. You know, you know, as I'm saying, this is a probability scale, you know, and I'm sure that, you know, this one is coherent and can be reused, you know. You know, it's a perfect strategy, because the strategy, this is a model of vectors in the reward, you know.

And, you know, that means the blue team, which are playing two defensively, are prone to losing the game, ultimately, you know. And they will lose, they've lost, you know, even before the game is started. Because the analytic framework for defence in the model, over the virtual one, over any attack, aggression, is imbalancing in the metric, you know.

One to one minute, you know. You know, the 0.1 here represents just basically 0.7 or 0.1% attack. 0.2 represent defence, or 0.6. 0.2 in the red probability of 0.6 in blue.

So basically, 0.7 minus 0.1 is 0.6. And this 0.6 is PR minus PB. And this is probability of the red over blue attack versus defence. It's more over the dimensions of attack, defence, offence, and information.

You know, these are just four dimensions that I also mean, at least these are dimensions. But they're no more than in the problem. And this is an imbalanced vector.

You know, this vector is a triangle sign. I don't know, you know. The change, the change in, this means the change in.

The change in, probability of R, you know, can be measured in real time. You know, that's what in the run time, in the robot, you know. It's a metric for defence coherence.

How good your dimensions are doing, you know. How good you are. Detect.

This is how opposite the strongest dimension. Neutralise. behaves as negative way to counter it.

Redire shift focus to adjacent dimensions. Output the bending strategy, vector, you know. You know, some sensitive.

So basically, it's a reverse 374. So basically, there isn't strong enough. Low appearance dimension.

That's your worst case scenario. You don't want that thing getting underfunded. Neutralise the weight, the weight to counter it.

So basically, now it's a more balanced view. That's a queen in the game. which redirects you from the statistic dimension and that's like the worst min-squares problem you're going to do it and the next one is like output the blended strategic vector you know now the blended strategic vector is like the executor of the vector in the road but you know it's like um so you get one detect 2D trace per redact you know and then that's new um forms a new um puts a new dimension there and gives you a probability score and that balanced probability score has been remeasured you know and that's after that's when the execution is taken place that's why it's real-time you know because you executed that in going new dimension but I mean dimension means that this variable of the dynamic thing but in a different context you know I have different objective you know you know uh using this you know same thing that's why the balance is measuring in real time okay now unicode security three now uh case study apply structural thing that's in cyber security the problem is multi encoding uh of the same attack vector bypass ratio files you know direct dot dot you know encoded basically the the encoded format the calculation does you know let me type this on a terminal this encoder format now the utf interpretation is problematic you know because you know these are these are these are traditional bypass filters you know because you know there's you know I can use this structural syntax to strike the problem multiple path encoding of the same attack vector bypass ratio filter so this I can put in this uh utf8 model of the encoder state and this will be you know I can I can uh encode this in utf8 you know you know but this was encoded into computer this is direct one this one's going to be a computer you know dot dot dot o2f you know every character but this utf calculation you know you know no this is an attack vector yes you know it's you know it's an attack vector you know because utf8 uh is not canonical you know you know this attack vector is the same as it can punch in the same you know or as a computer programme and it can bypass the root access dot dot you know if that is common direction or dot dot dot is like this one but this utf8 system doesn't understand uh doesn't don't think this is the encoding of the utf8 system uh which stands for unified uh unicode uh it's a unicode format of eight bytes of memory uh eight bytes eight bits to a one byte uh model uh what's called utf8 because it comes eight times eight it's got 64 bytes I think yes I believe so so basically I sort of wrote this concept you know I took a state different encoder I just path uh different encoders are just different paths to semantic states we collect them into still colour form the colour code form which is the simplest model that's used for the standard application of real world systems you know kind of code just means universal you know but it's the simplest form of the universal framework you know the kind of code form is the it's like um it's like you know ensuring that uh it's as long as it bugs a smoke reduction of states you know other states is chemical you know uh it's a feature it's a bug it's a feature it's not a bug you know you have to reduce all the all the code all the encoding you know one is to ensure that you cannot you cannot route out of your uh frame of records with any interface system provided by the user or the hacker state cia you know fly here a timer stop stop fly me fly so basically state side uh flash time that's what I'm saying using us in a unc uh uh gramme and three states recognition classical inputs are by code inside so map of the equivalent state in colour code form three apply for uh a single unified security policy you know the thing about this is like the recognition of the you cn your unicode um security uh characterisation of normalisation you know normalisation means like uh the the the smallest form so kind of code is just like the simplest one of course the normalise like one system can be one system but normalised form is called code form too uh and then this uh uh unicode uh structure uh uh uh colour code normalisation algorithms a three-phase pipeline the result of this