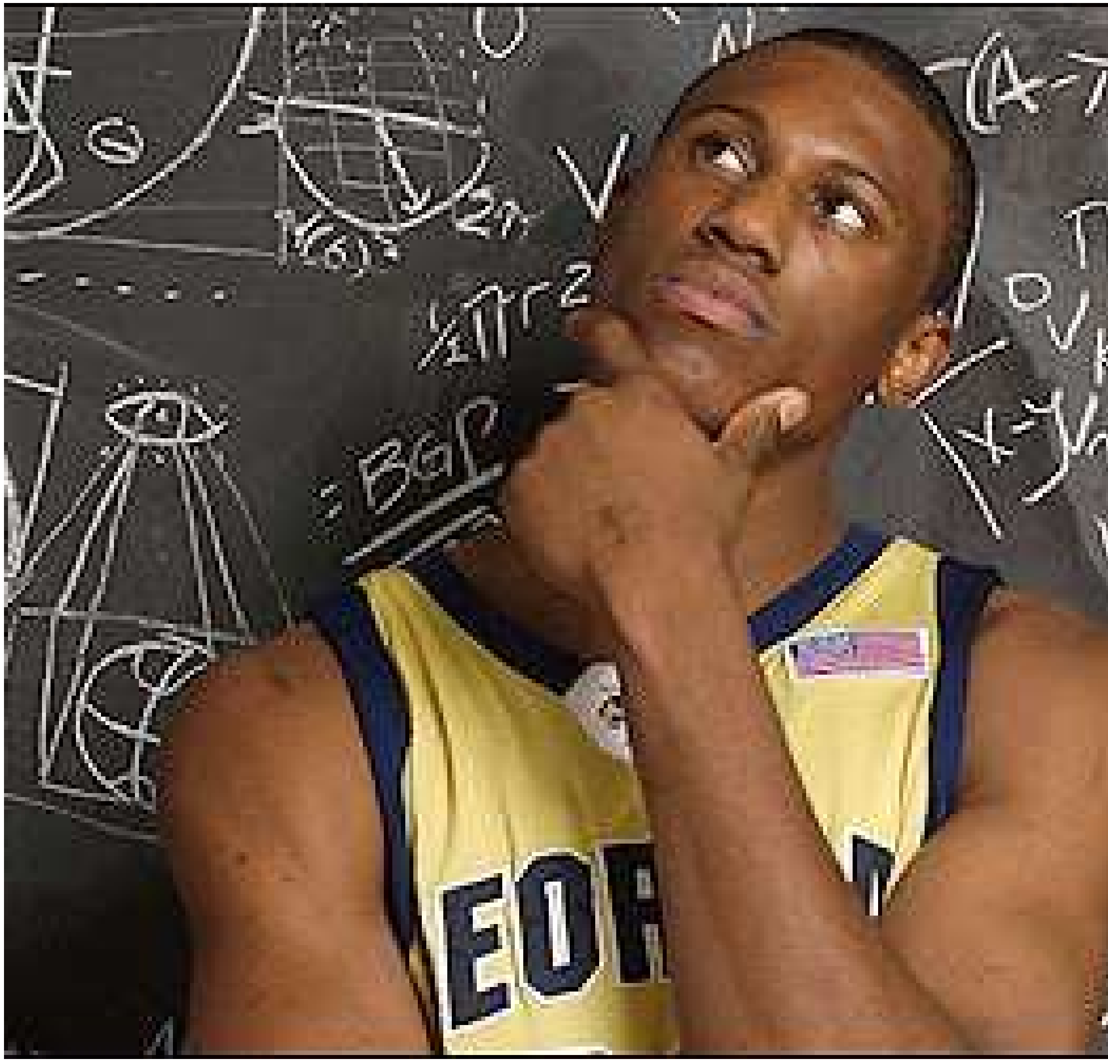
Multi-agent learning

Satisficing play

Gerard Vreeswijk, Intelligent Software Systems, Computer Science Department, Faculty of Sciences, Utrecht University, The Netherlands.

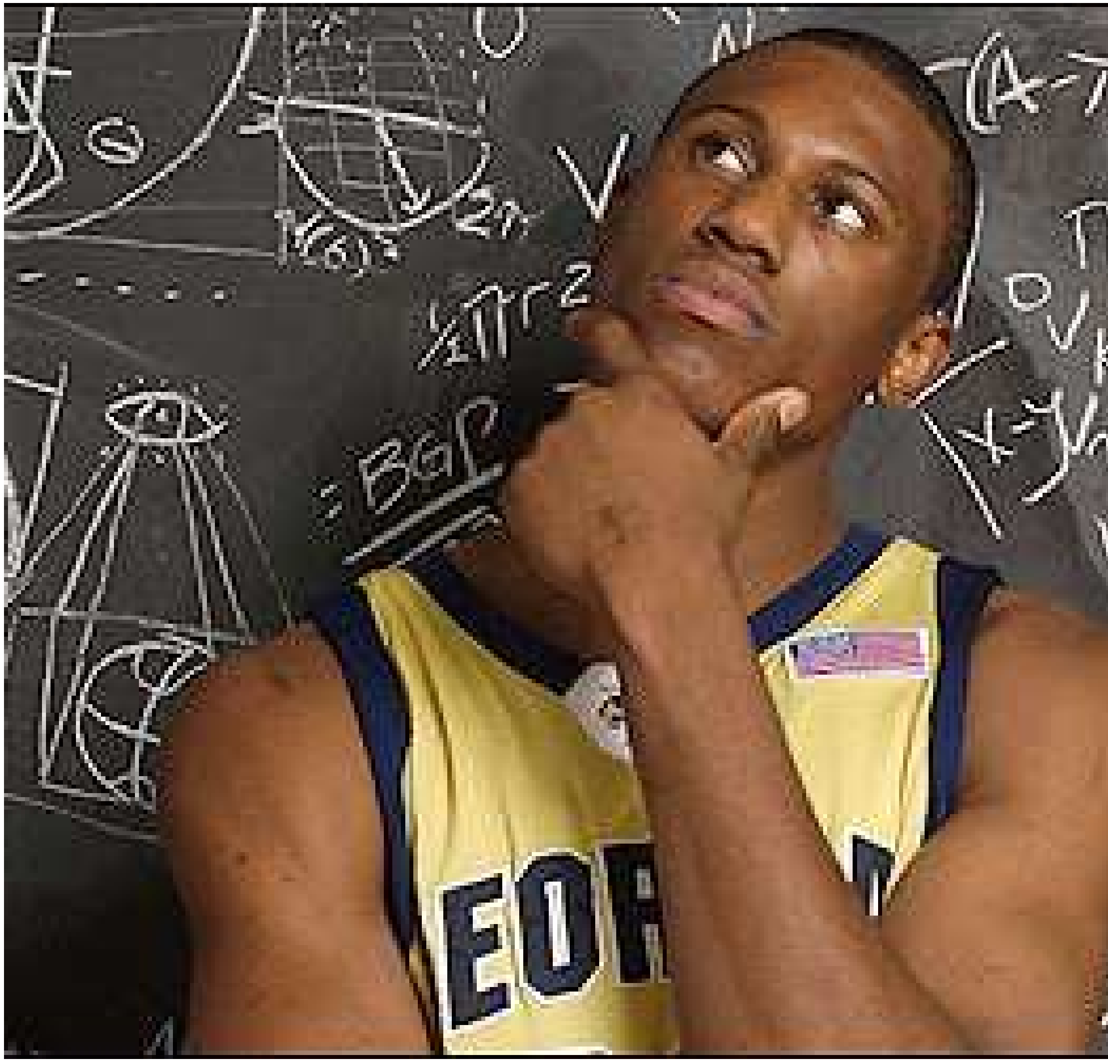
Tuesday 16th June, 2020

Assumptions in game playing



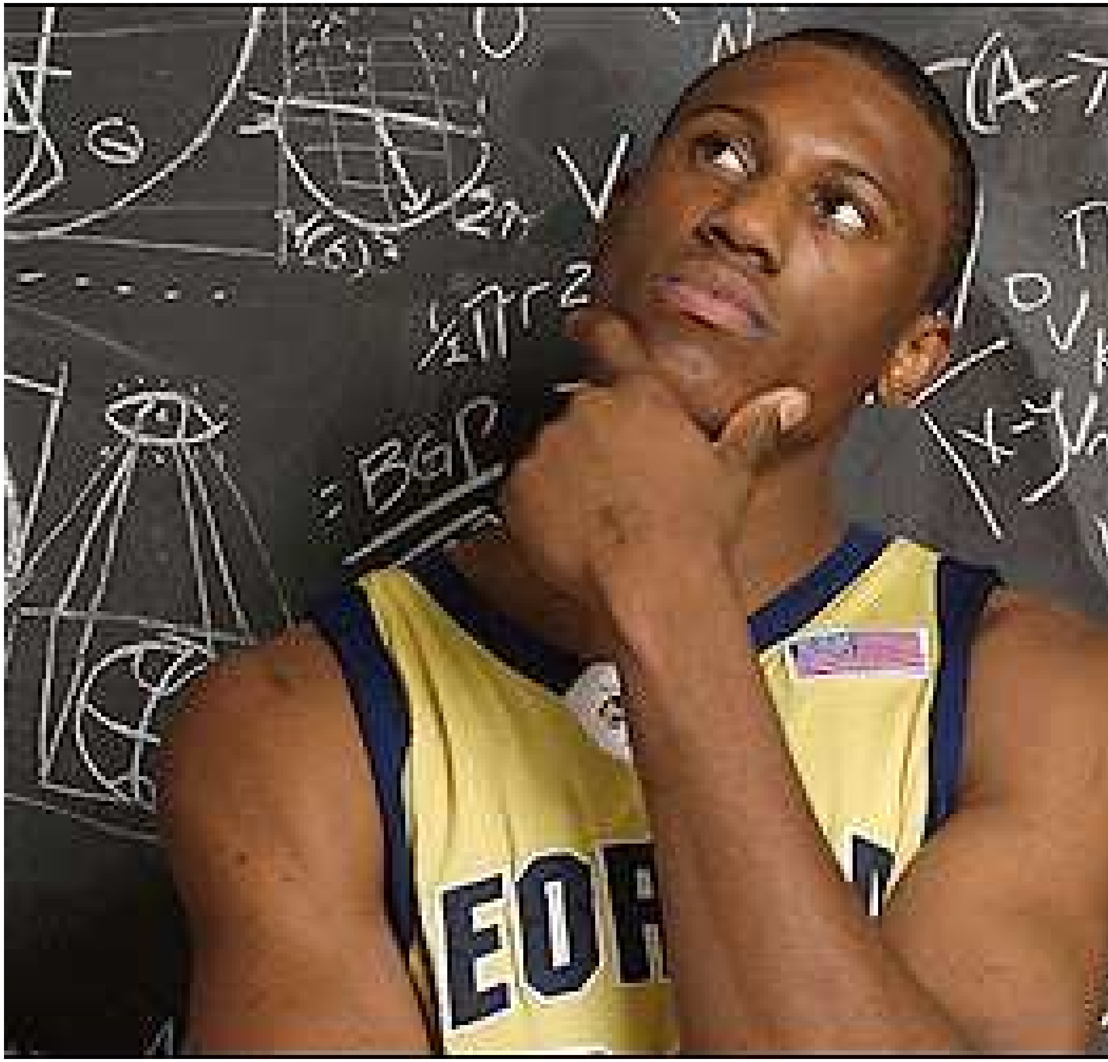
Assumptions in game playing

- Players know the the structure of the game, such as:



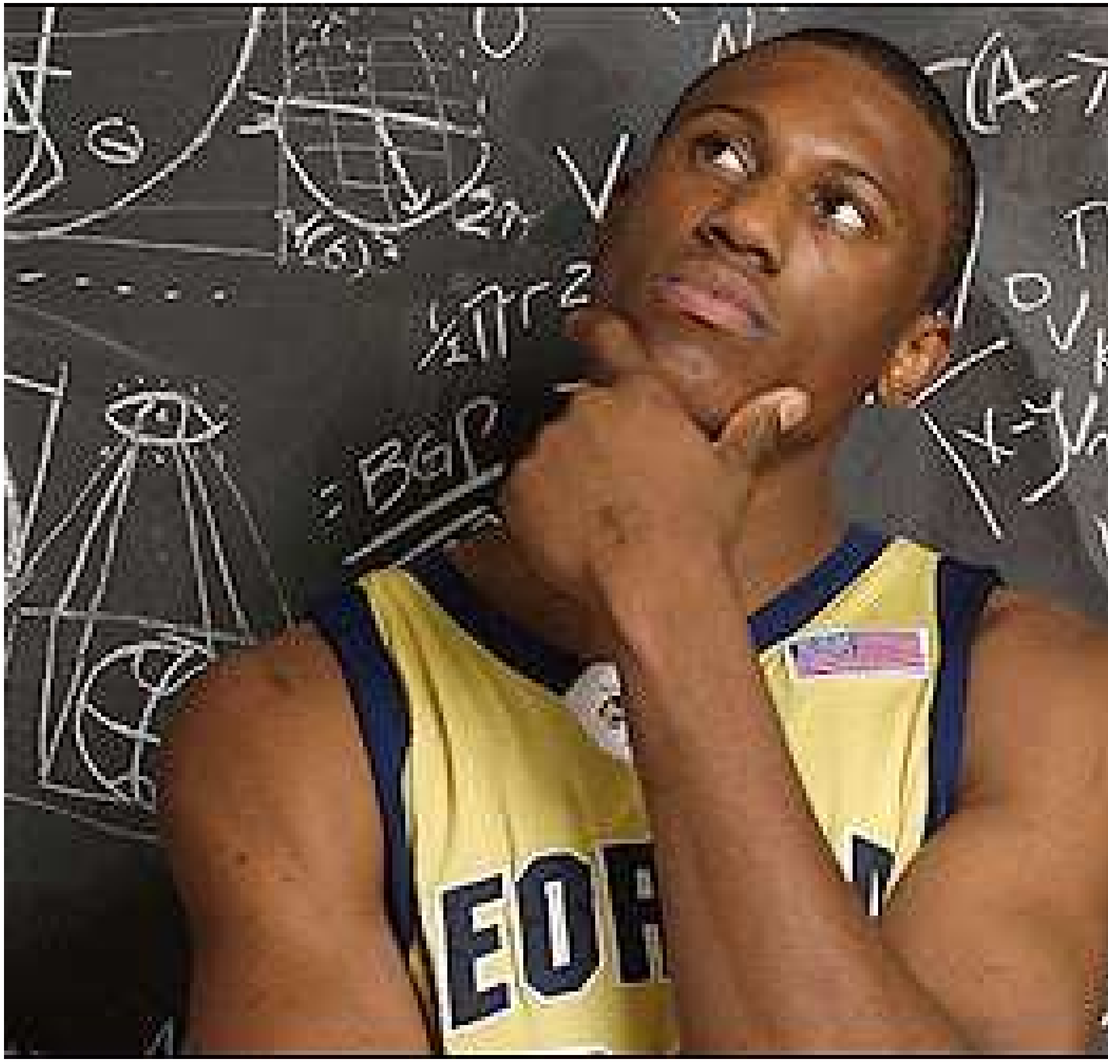
Assumptions in game playing

- Players know the the structure of the game, such as:
  - Other players.



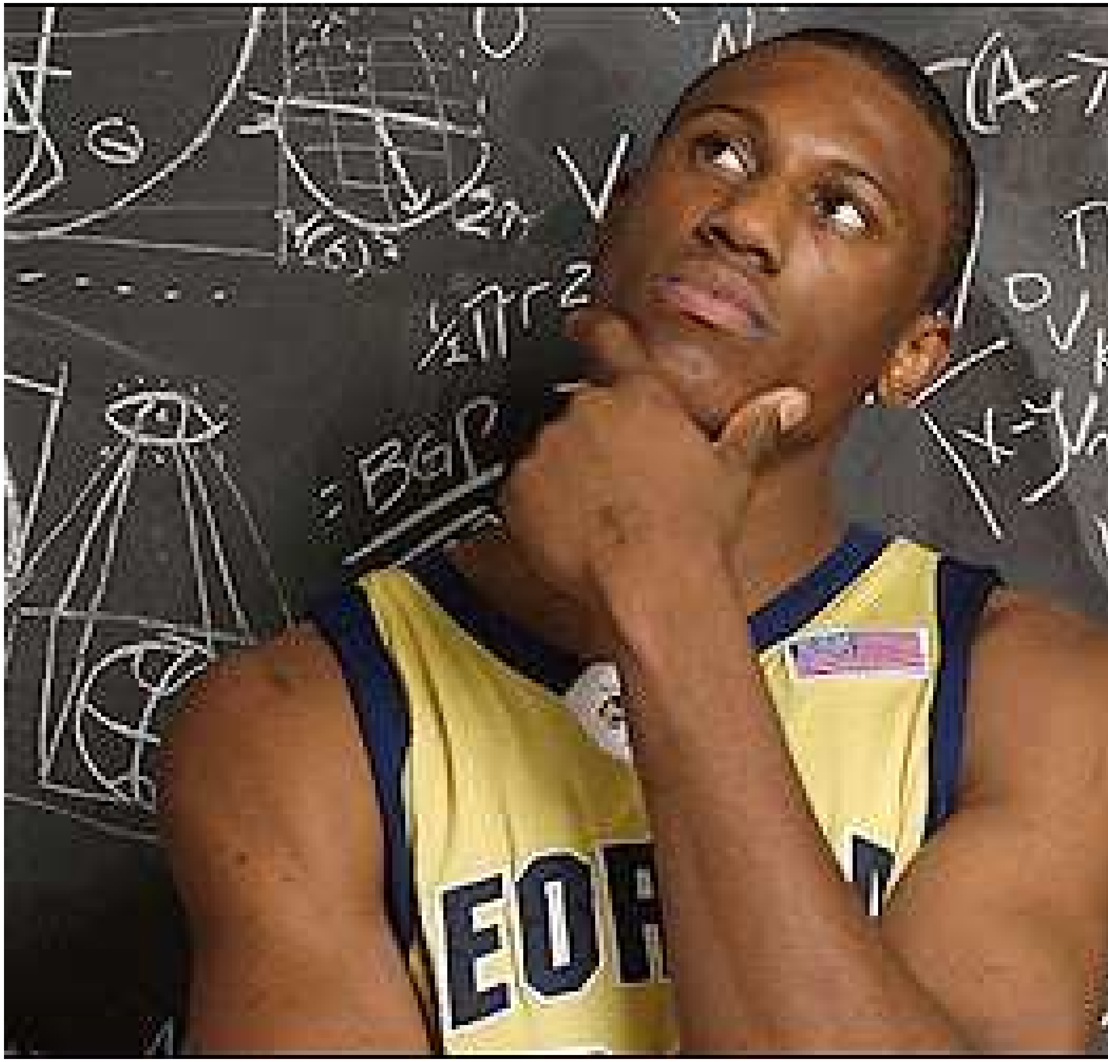
Assumptions in game playing

- Players know the the structure of the game, such as:
  - Other players.
  - Other player’s possible actions.



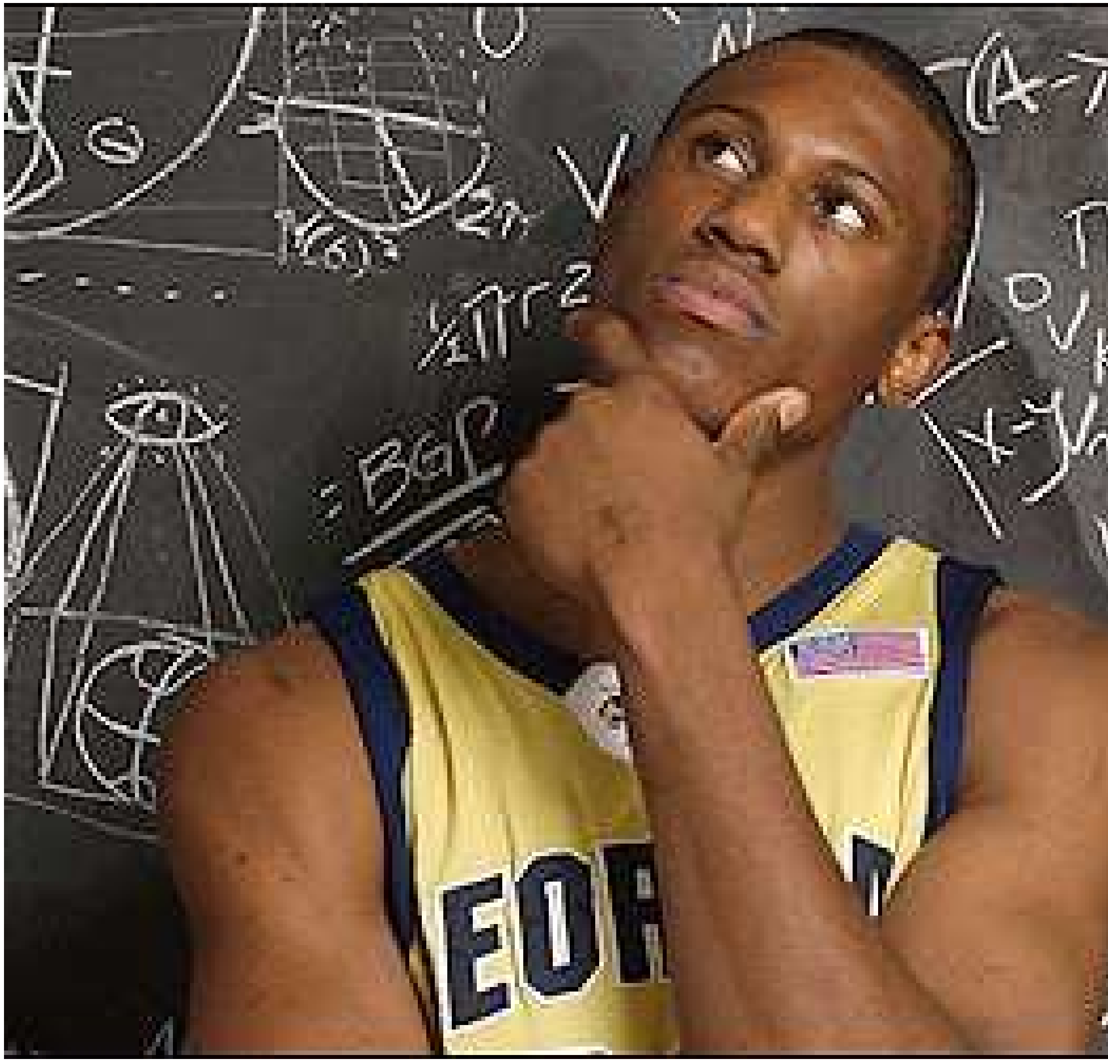
Assumptions in game playing

- Players know the the structure of the game, such as:
  - Other players.
  - Other player's possible actions.
  - Relationship between actions and payoffs.



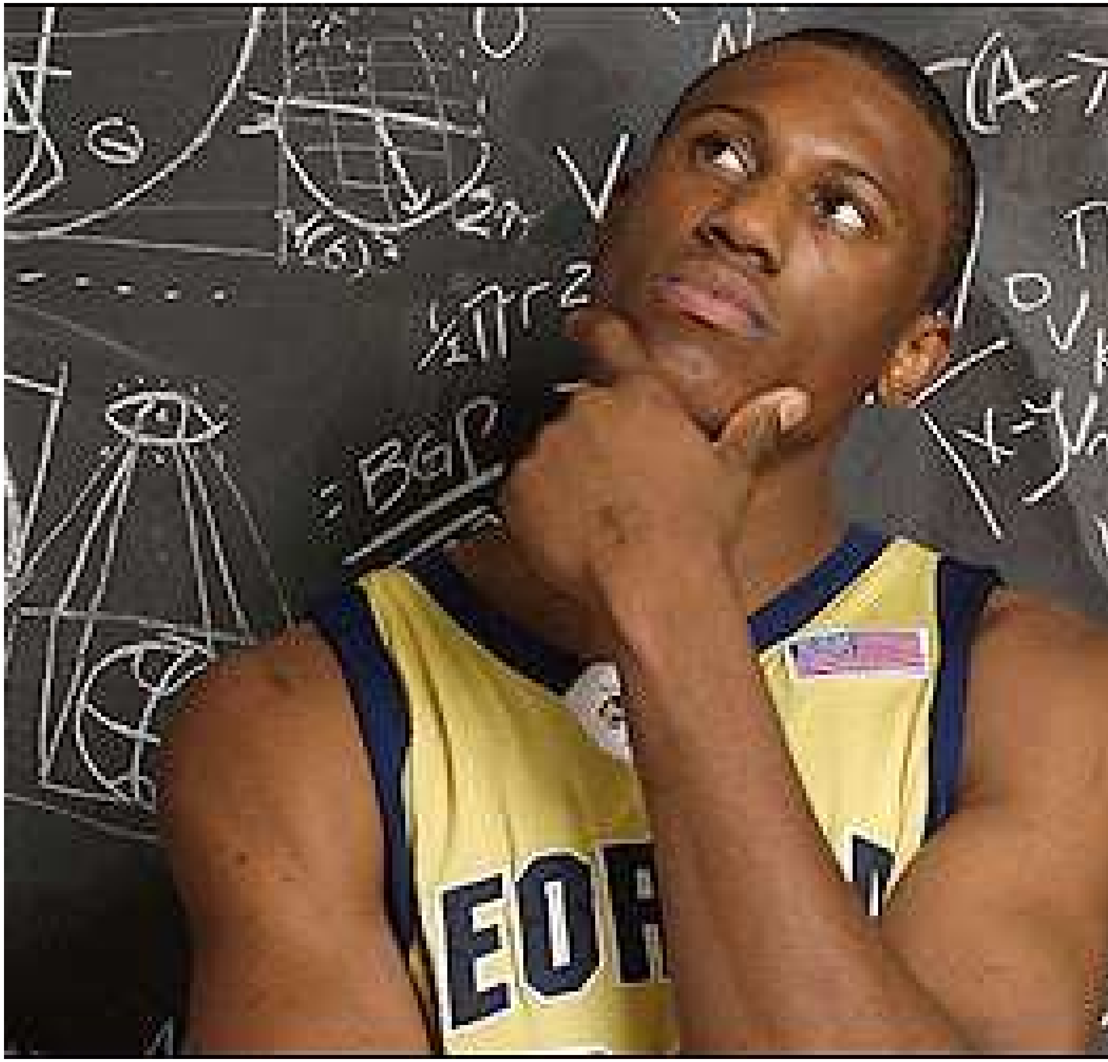
Assumptions in game playing

- Players know the the structure of the game, such as:
  - Other players.
  - Other player's possible actions.
  - Relationship between actions and payoffs.
- Players can observe other player's actions.



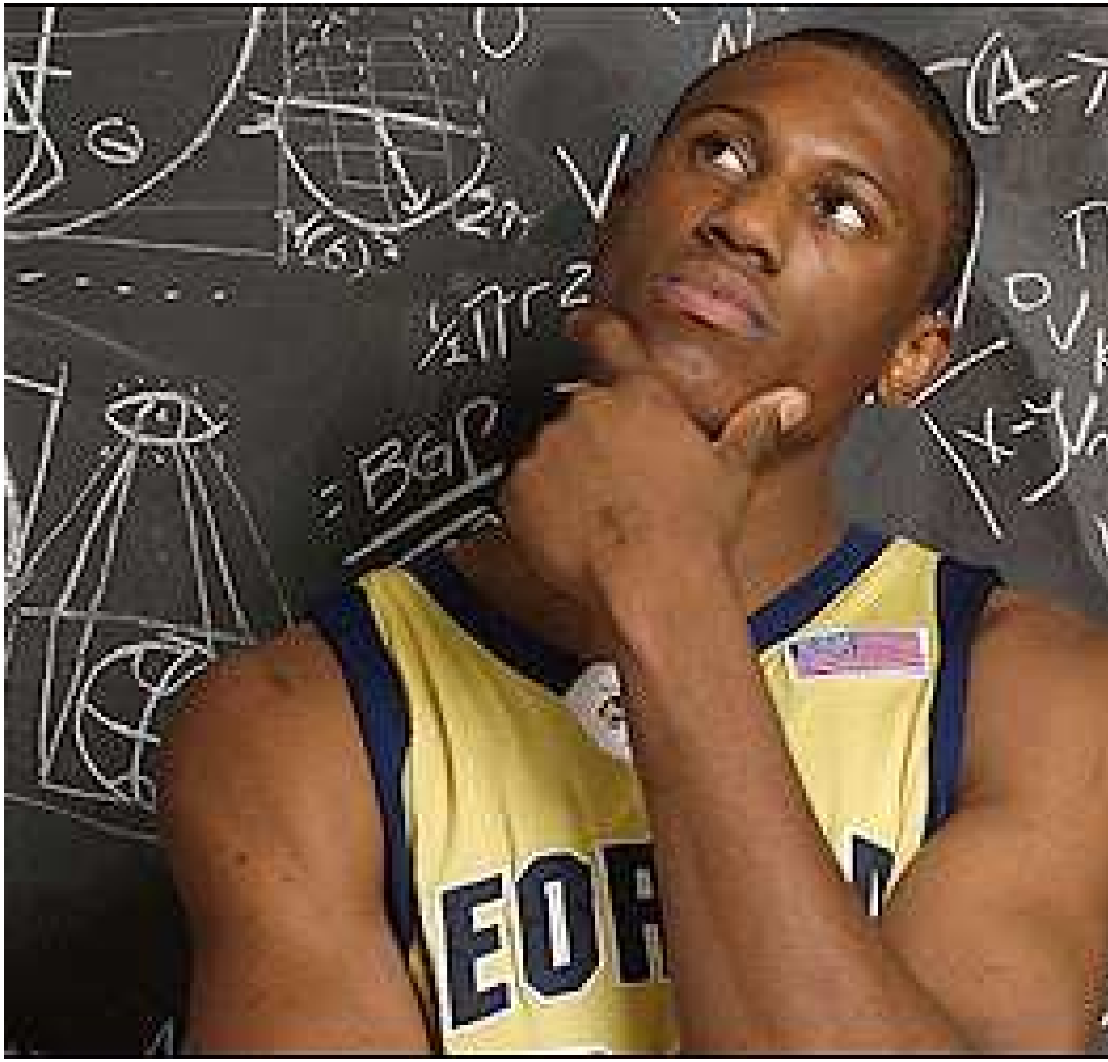
Assumptions in game playing

- Players know the the structure of the game, such as:
  - Other players.
  - Other player's possible actions.
  - Relationship between actions and payoffs.
- Players can observe other player's actions.
- . . . other player's payoffs.



Assumptions in game playing

- Players know the the structure of the game, such as:
  - Other players.
  - Other player's possible actions.
  - Relationship between actions and payoffs.
- Players can observe other player's actions.
- …other player's payoffs.
- Players are aware that they are in a game.



What if none of these assumptions holds?

What if none of these assumptions holds?

- Players don't know the structure of the game.

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.
- Players can't observe other player's payoffs.

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.
- Players can't observe other player's payoffs.
- Players aren't aware that they are in a game.

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.
- Players can't observe other player's payoffs.
- Players aren't aware that they are in a game.

This takes the problem out of game theory into machine learning.

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.
- Players can't observe other player's payoffs.
- Players aren't aware that they are in a game.

This takes the problem out of game theory into machine learning.

What can we do?

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.
- Players can't observe other player's payoffs.
- Players aren't aware that they are in a game.

This takes the problem out of game theory into machine learning.

What can we do?

- Reinforcement learning.

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.
- Players can't observe other player's payoffs.
- Players aren't aware that they are in a game.

This takes the problem out of game theory into machine learning.

What can we do?

- Reinforcement learning.

  Disadvantages:

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.
- Players can't observe other player's payoffs.
- Players aren't aware that they are in a game.

This takes the problem out of game theory into machine learning.

What can we do?

- Reinforcement learning.

  Disadvantages:

  - No reference to past average payoffs.

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.
- Players can't observe other player's payoffs.
- Players aren't aware that they are in a game.

This takes the problem out of game theory into machine learning.

What can we do?

- Reinforcement learning.

  Disadvantages:

  - No reference to past average payoffs.
  - Difficult theory.

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.
- Players can't observe other player's payoffs.
- Players aren't aware that they are in a game.

This takes the problem out of game theory into machine learning.

What can we do?

- Reinforcement learning.

  Disadvantages:

  - No reference to past average payoffs.
  - Difficult theory.

Alternative:

What if none of these assumptions holds?

- Players don't know the structure of the game.
  - Players don't know who's playing.
  - Players don't know the arsenal of other players.
- Players can't observe other player's actions.
- Players can't observe other player's payoffs.
- Players aren't aware that they are in a game.

This takes the problem out of game theory into machine learning.

What can we do?

- Reinforcement learning.

  Disadvantages:

  - No reference to past average payoffs.
  - Difficult theory.

Alternative:

- Satisficing learning.

Herbert A. Simon on maximising vs. satisficing

Herbert A. Simon on maximising vs. satisficing

Herbert A. Simon on maximising vs. satisficing

Herbert A. Simon on maximising vs. satisficing

"A decision maker who chooses the best available alternative according to some criteria is said to optimise; one who chooses an alternative that meets or exceeds specified criteria, but that is not guaranteed to be either unique or in any sense the best, is said to satisfice ."[a]

[a] H. Simon "Models of bounded rationality" in: *Empirically grounded economic reasons*, Vol. 3. MIT Press, 1997.

Herbert A. Simon on maximising vs. satisficing

"A decision maker who chooses the best available alternative according to some criteria is said to optimise; one who chooses an alternative that meets or exceeds specified criteria, but that is not guaranteed to be either unique or in any sense the best, is said to satisfice ."[a]

[a] H. Simon "Models of bounded rationality" in: *Empirically grounded economic reasons*, Vol. 3. MIT Press, 1997.

"Decision makers can satisfice either by finding optimum solutions for a simplified world, or by finding satisfactory solutions for a more realistic world.

Herbert A. Simon on maximising vs. satisficing

"A decision maker who chooses the best available alternative according to some criteria is said to optimise; one who chooses an alternative that meets or exceeds specified criteria, but that is not guaranteed to be either unique or in any sense the best, is said to satisfice ."[a]

[a] H. Simon "Models of bounded rationality" in: *Empirically grounded economic reasons*, Vol. 3. MIT Press, 1997.

"Decision makers can satisfice either by finding optimum solutions for a simplified world, or by finding satisfactory solutions for a more realistic world. Neither approach, in general, dominates the other, and both have continued to co-exist in the world of management science."

Herbert A. Simon on maximising vs. satisficing

"A decision maker who chooses the best available alternative according to some criteria is said to optimise; one who chooses an alternative that meets or exceeds specified criteria, but that is not guaranteed to be either unique or in any sense the best, is said to satisfice ."[a]

[a] H. Simon "Models of bounded rationality" in: *Empirically grounded economic reasons*, Vol. 3. MIT Press, 1997.

"Decision makers can satisfice either by finding optimum solutions for a simplified world, or by finding satisfactory solutions for a more realistic world. Neither approach, in general, dominates the other, and both have continued to co-exist in the world of management science."[a]

[a] H. Simon "Rational decision making in business organizations" in: The American Economic Review, Vol. 69(4), pp. 493-513.

Karandikar *et al.*'s algorithm for satisficing play (1989)

Satisficing algorithm

Satisficing algorithm

- At any time, t, the agent's state is a tuple (A_t, α_t).

Satisficing algorithm

- At any time, t, the agent's state is a tuple (A_t, α_t).
  - A_t is the current action.

Satisficing algorithm

- At any time, t, the agent's state is a tuple (A_t, α_t).
  - A_t is the current action.
  - α_t is the current aspiration level.

Satisficing algorithm

- At any time, t, the agent's state is a tuple (A_t, α_t).
  - A_t is the current action.
  - α_t is the current aspiration level. Updated as

$$\alpha_{t+1} =_{Def} \lambda\alpha_t + (1-\lambda)\pi_t,$$

Satisficing algorithm

- At any time, t, the agent's state is a tuple (A_t, α_t).
  - A_t is the current action.
  - α_t is the current aspiration level. Updated as

$$\alpha_{t+1} =_{Def} \lambda\alpha_t + (1-\lambda)\pi_t,$$

  where λ is the persistence rate

Satisficing algorithm

- At any time, t, the agent's state is a tuple (A_t, α_t).
  - A_t is the current action.
  - α_t is the current aspiration level. Updated as

$$\alpha_{t+1} =_{Def} \lambda \alpha_t + (1 - \lambda)\pi_t,$$

where λ is the persistence rate, and π_t is the payoff in round t.

Satisficing algorithm

- At any time, t, the agent's state is a tuple (A_t, α_t).
  - A_t is the current action.
  - α_t is the current aspiration level. Updated as

$$\alpha_{t+1} =_{Def} \lambda \alpha_t + (1 - \lambda)\pi_t,$$

    where λ is the persistence rate, and π_t is the payoff in round t.
  - It is up to the programmer to choose an initial action A_0, and an initial aspiration level α_0.

Satisficing algorithm

- At any time, t, the agent's state is a tuple (A_t, α_t).
  - A_t is the current action.
  - α_t is the current aspiration level. Updated as

    $$\alpha_{t+1} =_{Def} \lambda\alpha_t + (1-\lambda)\pi_t,$$

    where λ is the persistence rate, and π_t is the payoff in round t.
  - It is up to the programmer to choose an initial action A_0, and an initial aspiration level α_0.
- Satisficing algorithm:

$$A_{t+1} = \begin{cases} A_t & \text{if } \pi_t \geq \alpha_t, \\ \text{any other action} & \text{else.} \end{cases}$$

Satisficing algorithm

- At any time, t, the agent's state is a tuple (A_t, α_t).
  - A_t is the current action.
  - α_t is the current aspiration level. Updated as

$$\alpha_{t+1} =_{Def} \lambda \alpha_t + (1 - \lambda)\pi_t,$$

    where λ is the persistence rate, and π_t is the payoff in round t.
  - It is up to the programmer to choose an initial action A_0, and an initial aspiration level α_0.
- Satisficing algorithm:

$$A_{t+1} = \begin{cases} A_t & \text{if } \pi_t \geq \alpha_t, \\ \text{any other action} & \text{else.} \end{cases}$$

  Also works if "any other action" is replaced by "any action".

Example of satisficing play

Game: prisoner's dilemma.

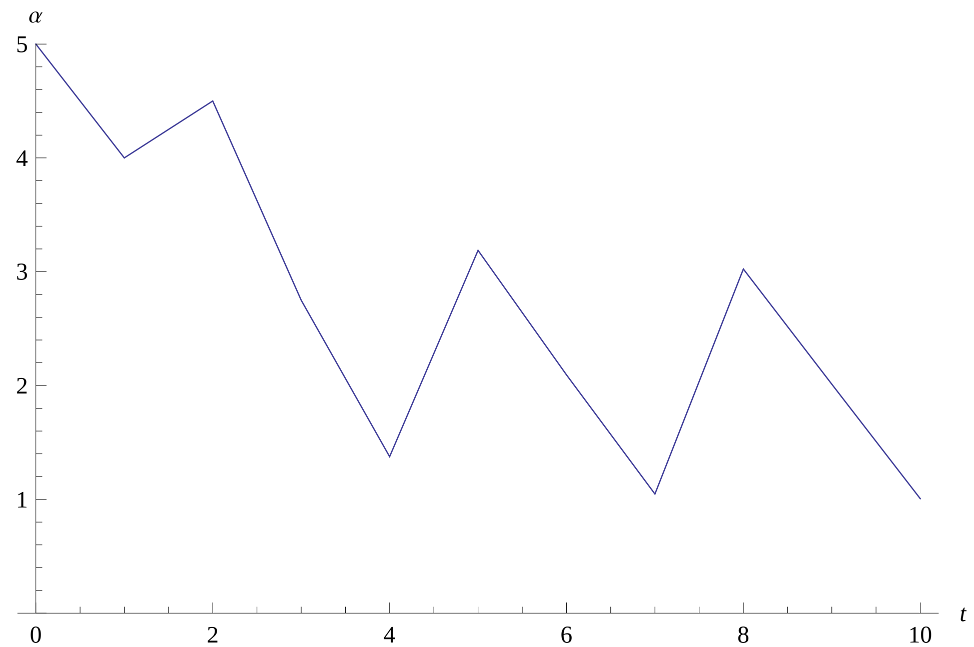
Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t

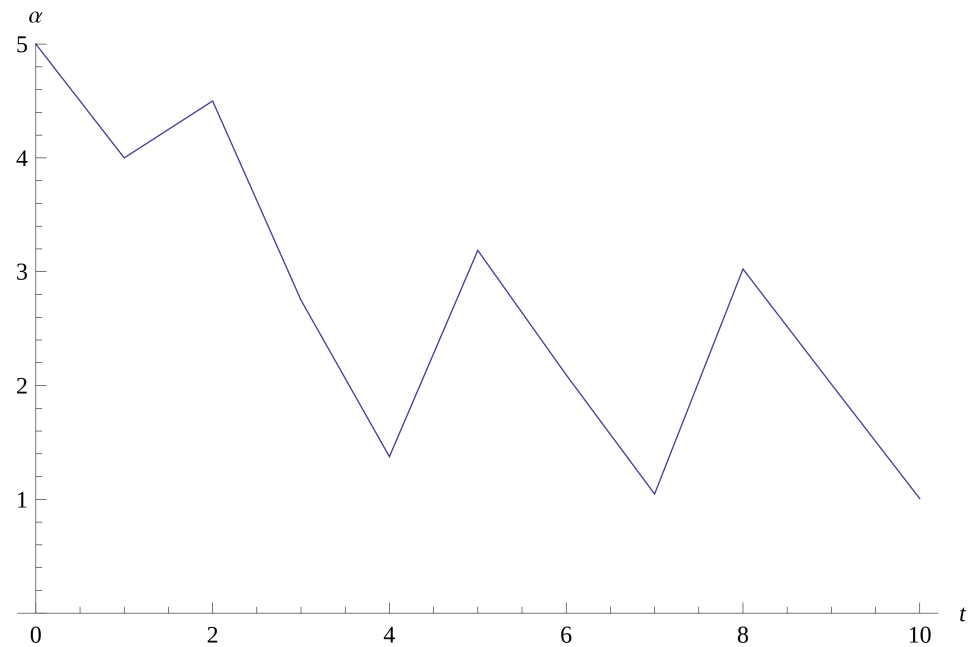


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t

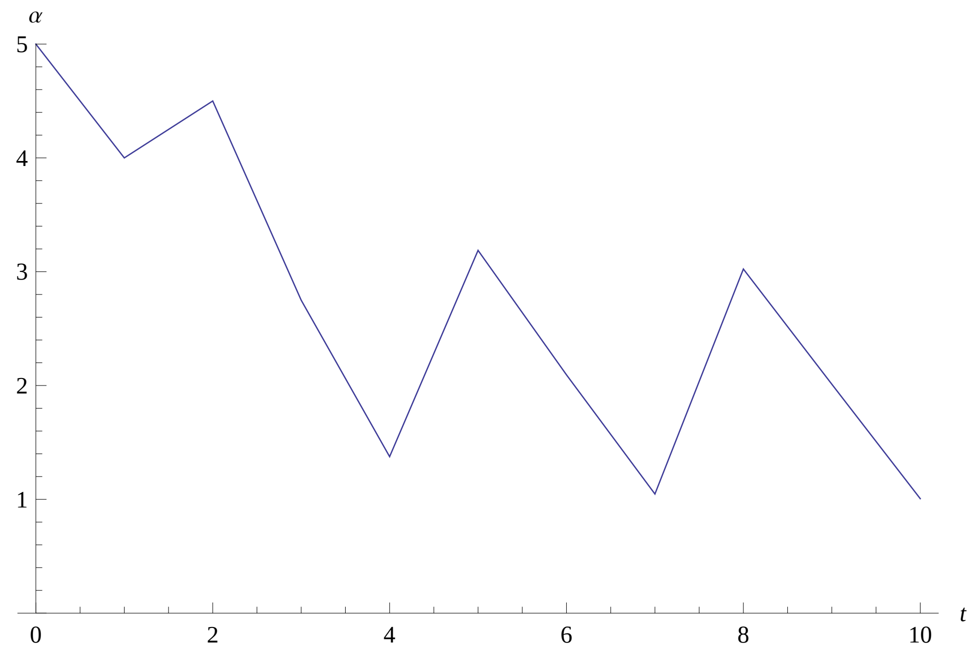


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0				

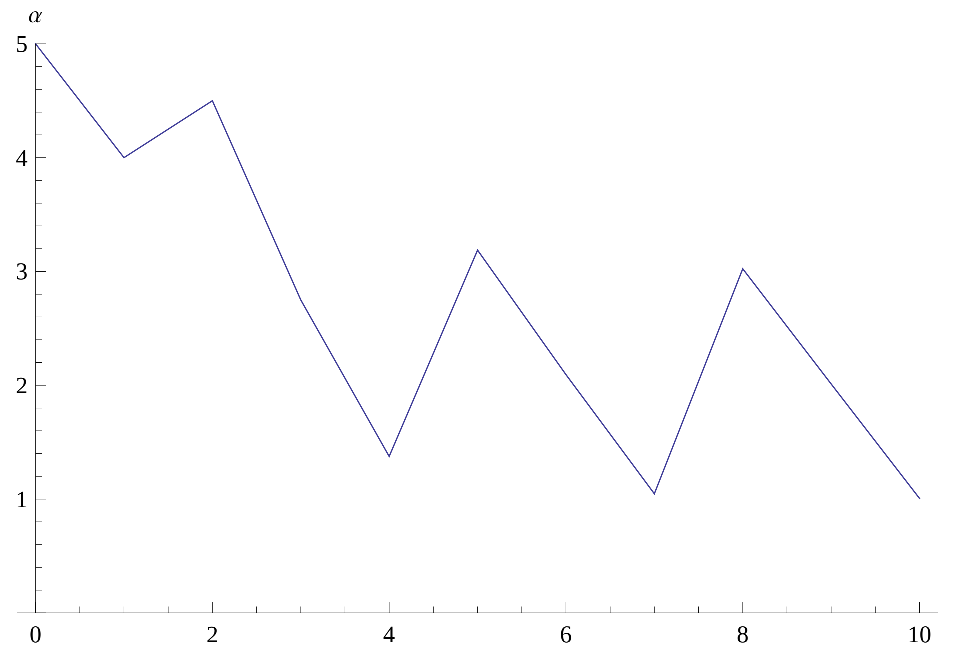


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C			

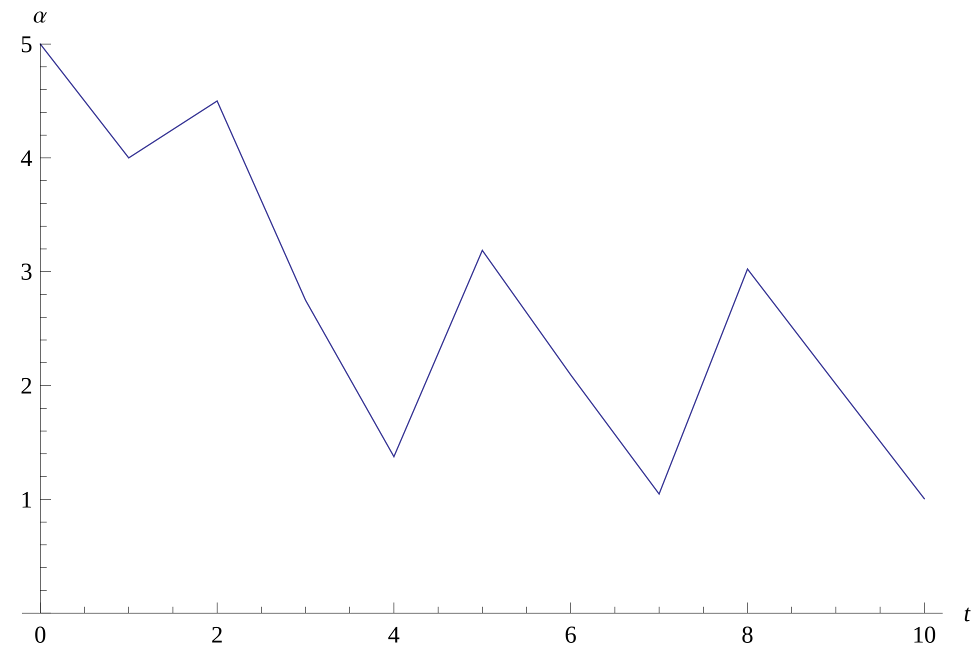


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C		

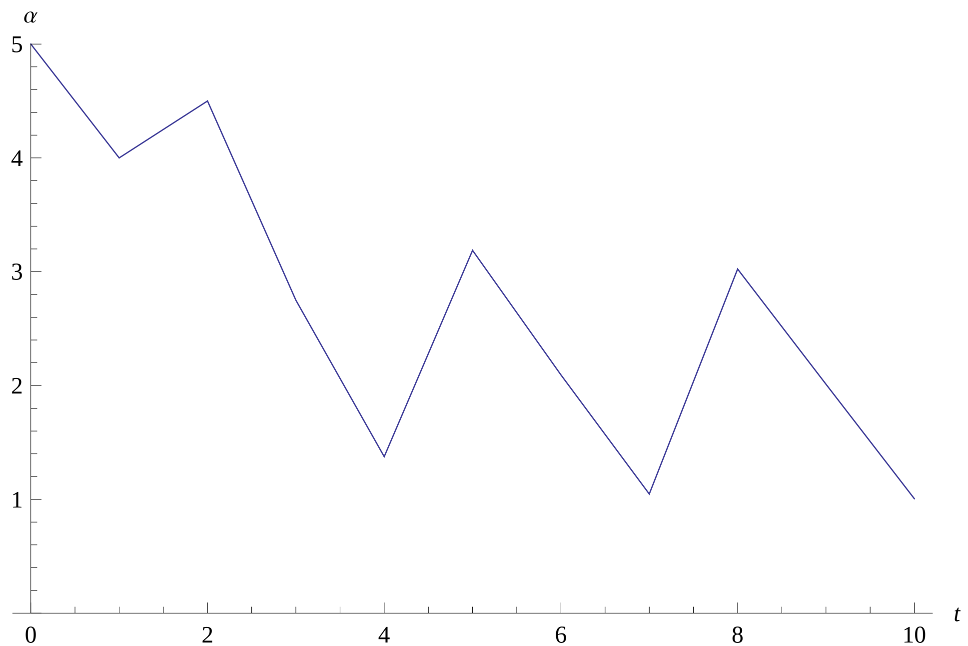


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	

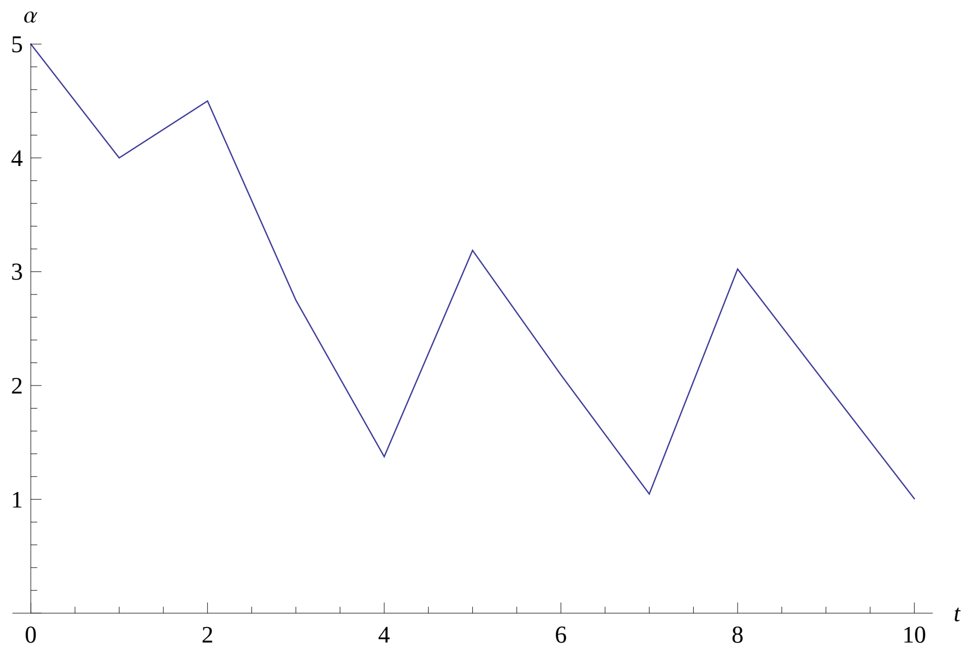


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5

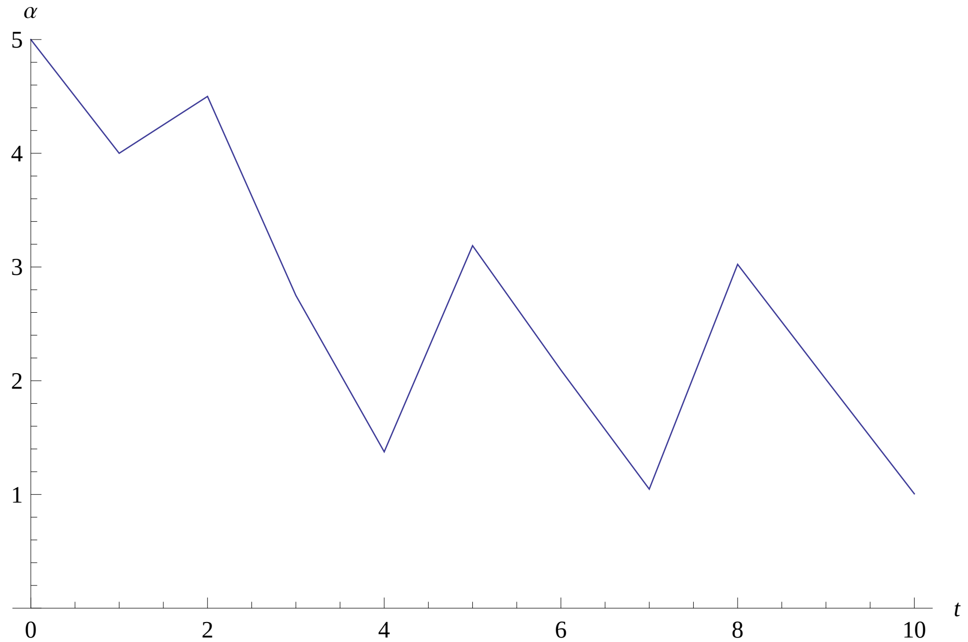


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1				

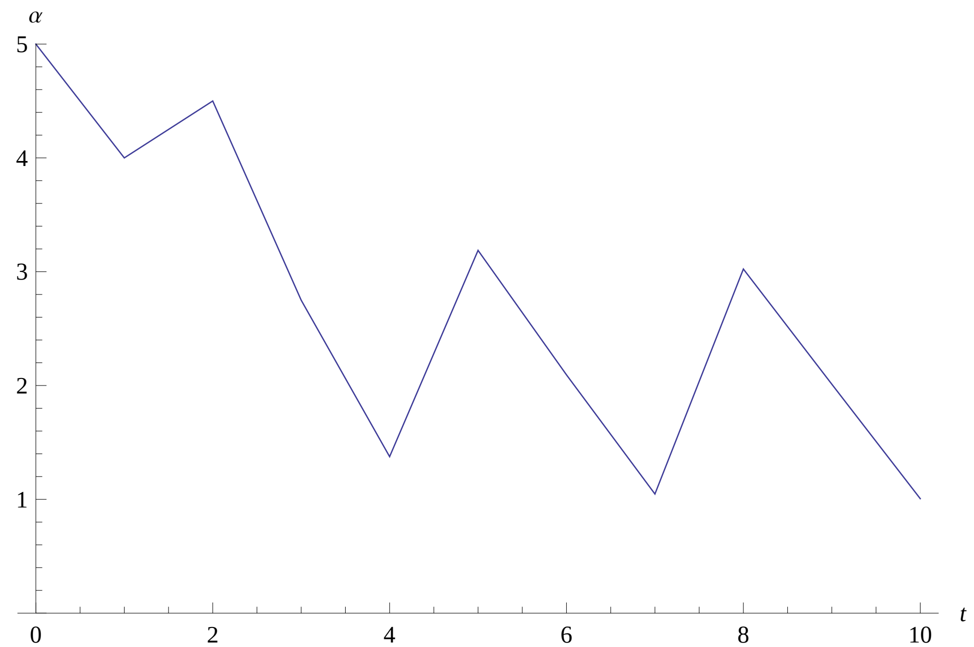


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C			

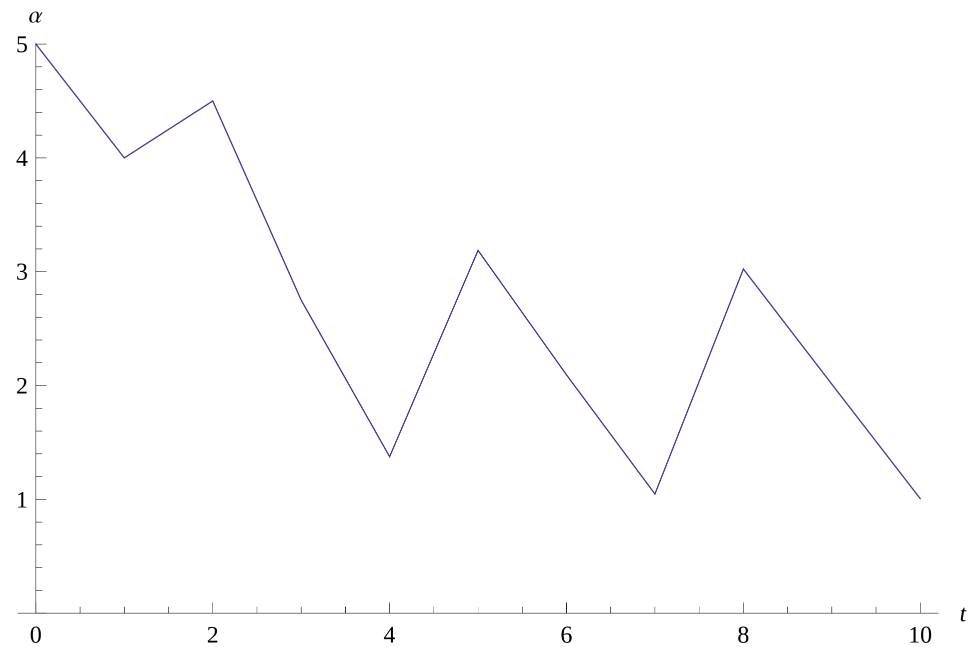


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**		

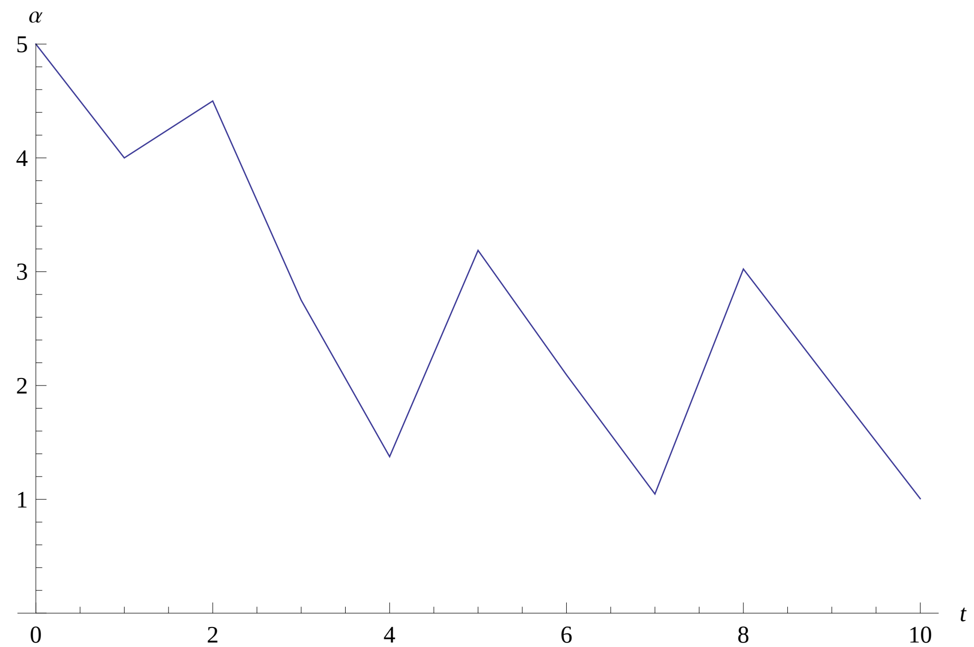


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	

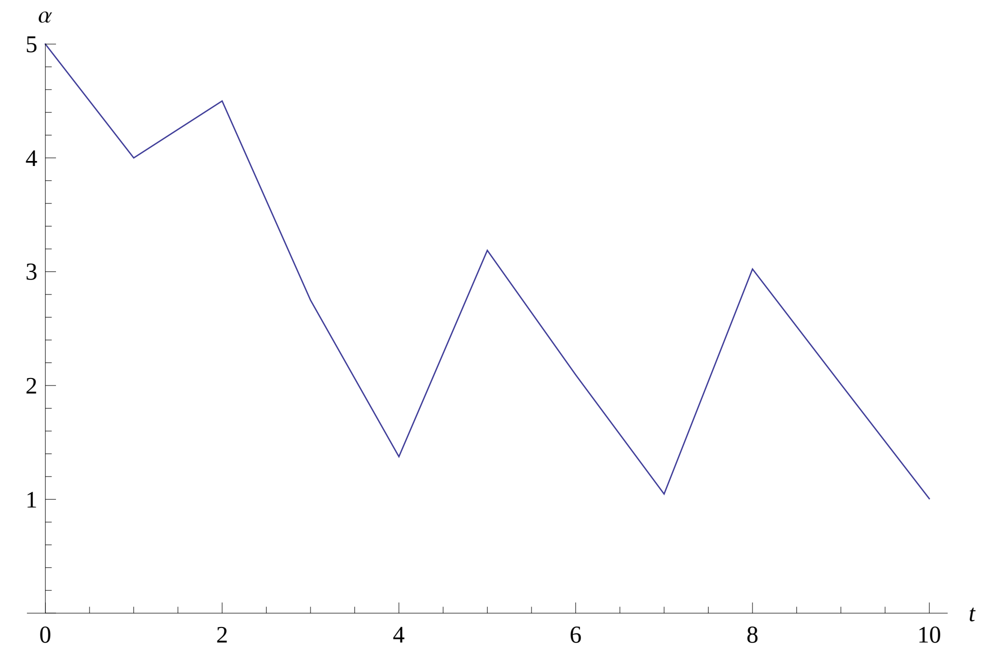


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4

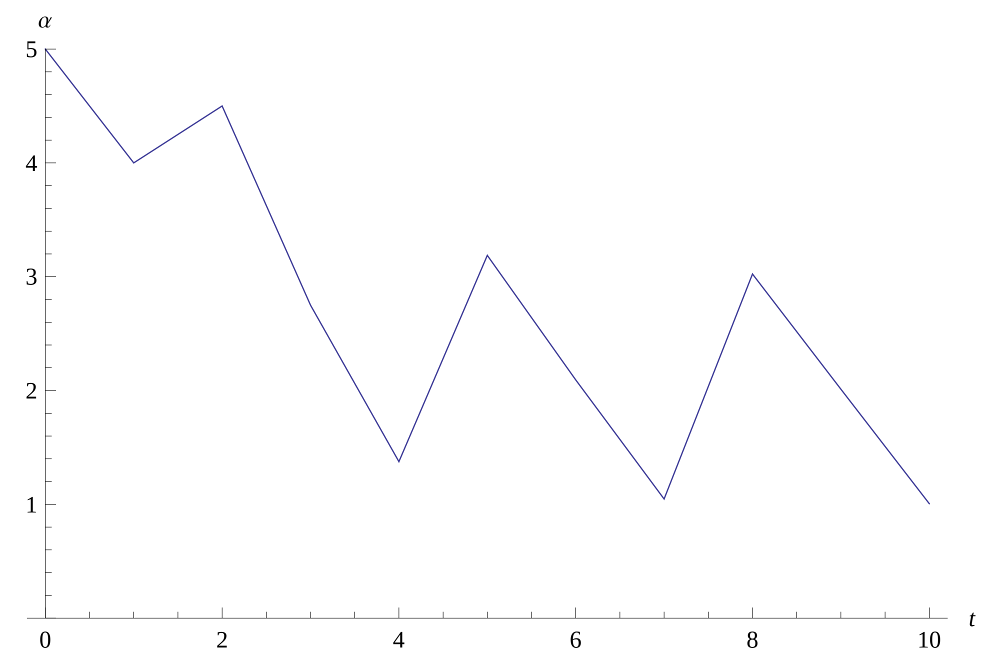


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2				

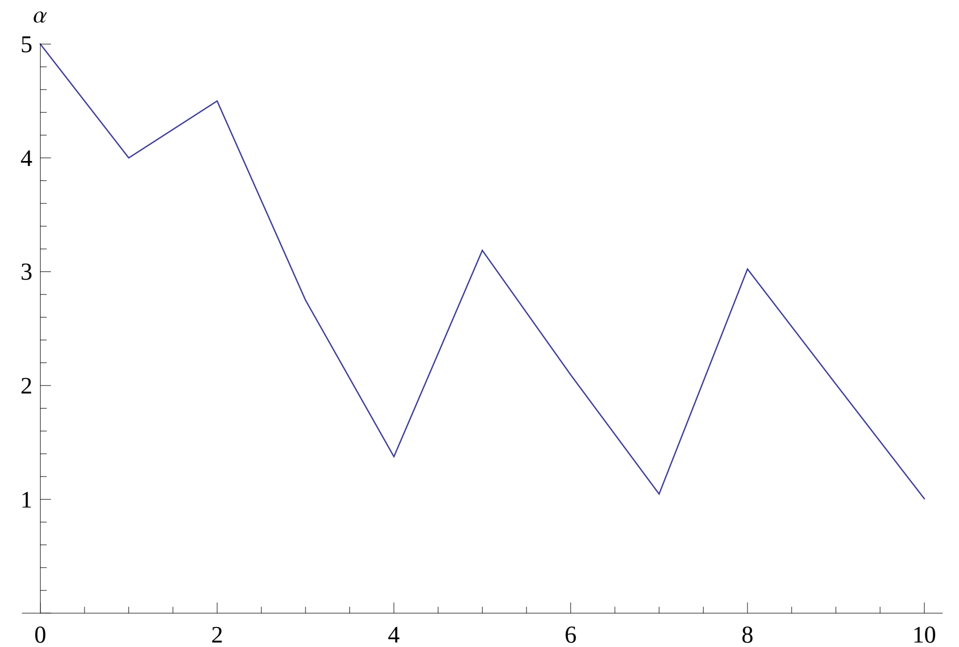


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D			

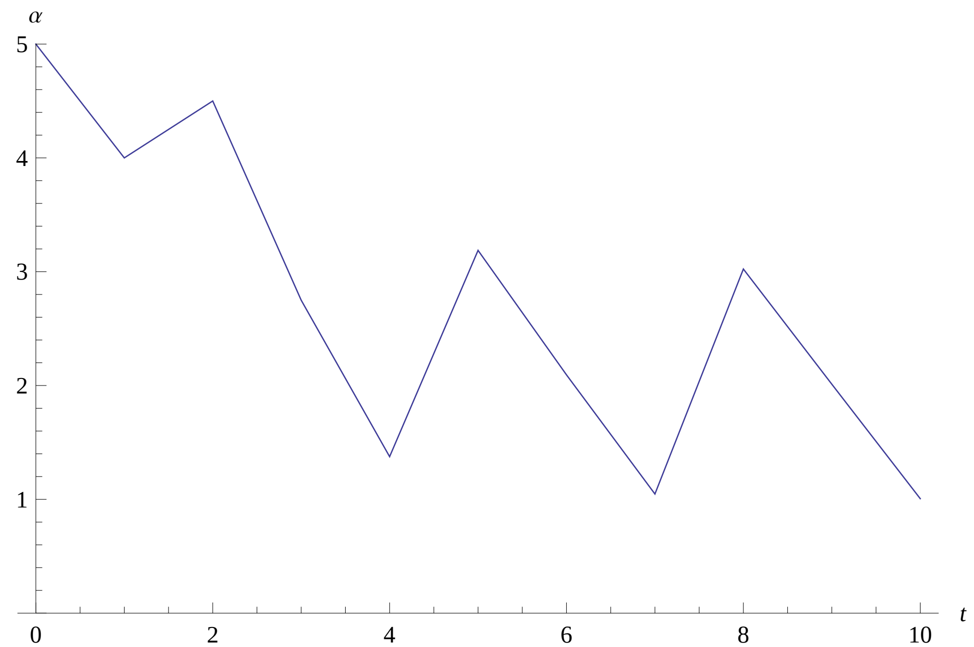


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D		

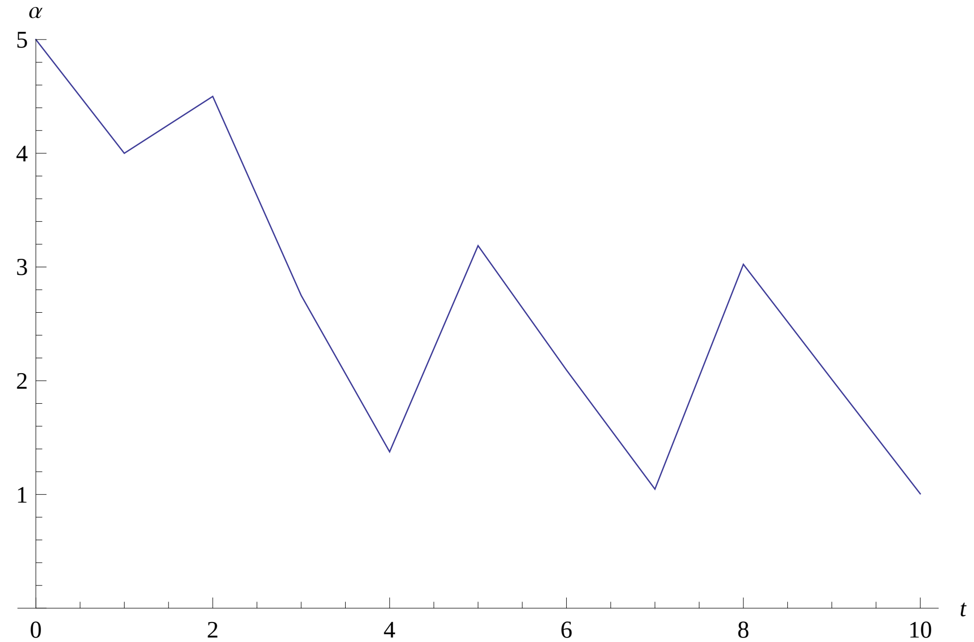


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	

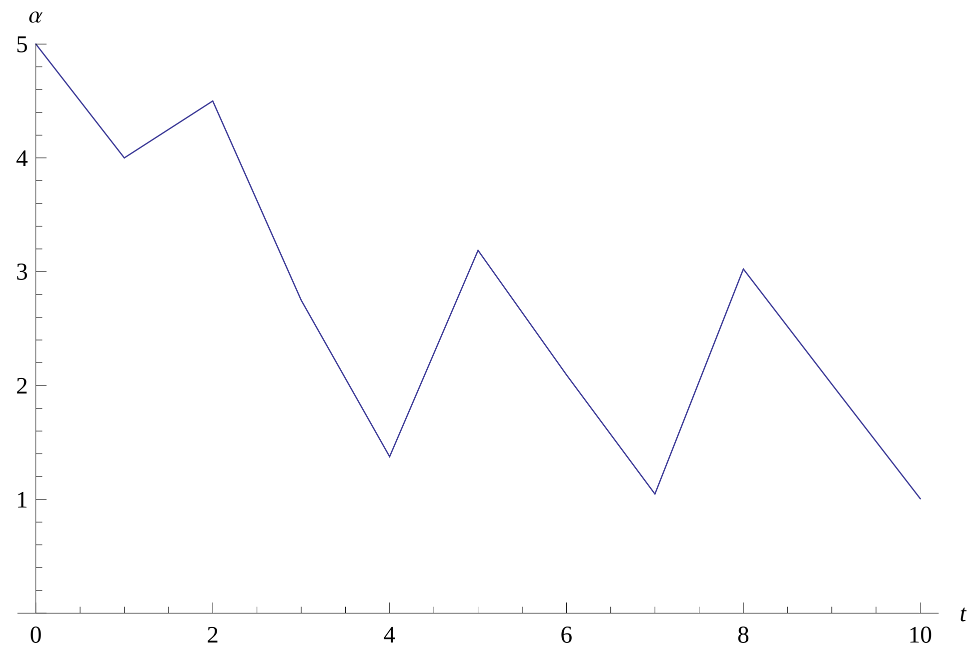


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5

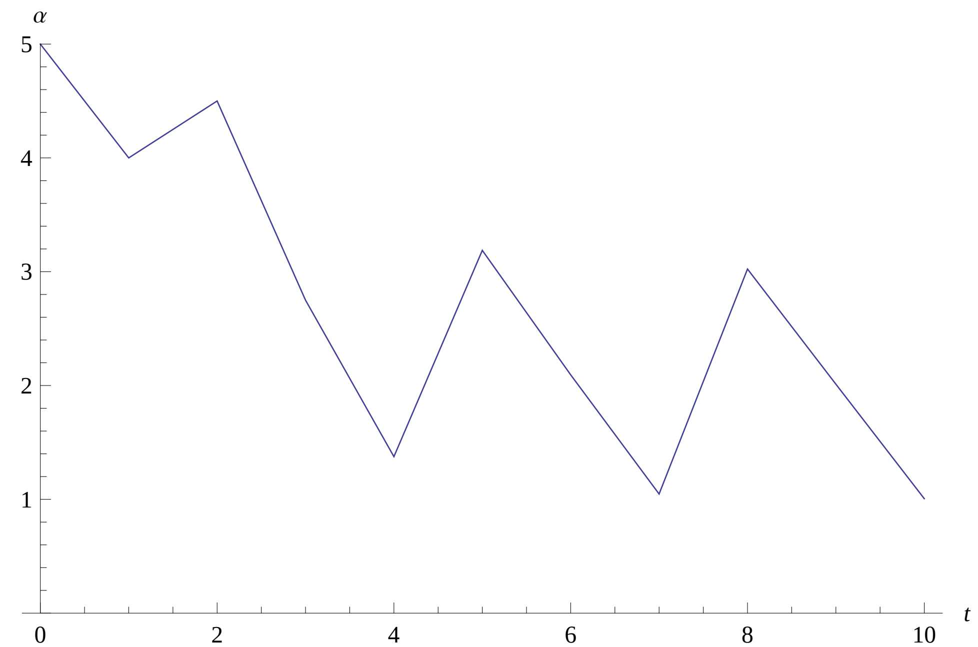


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3				

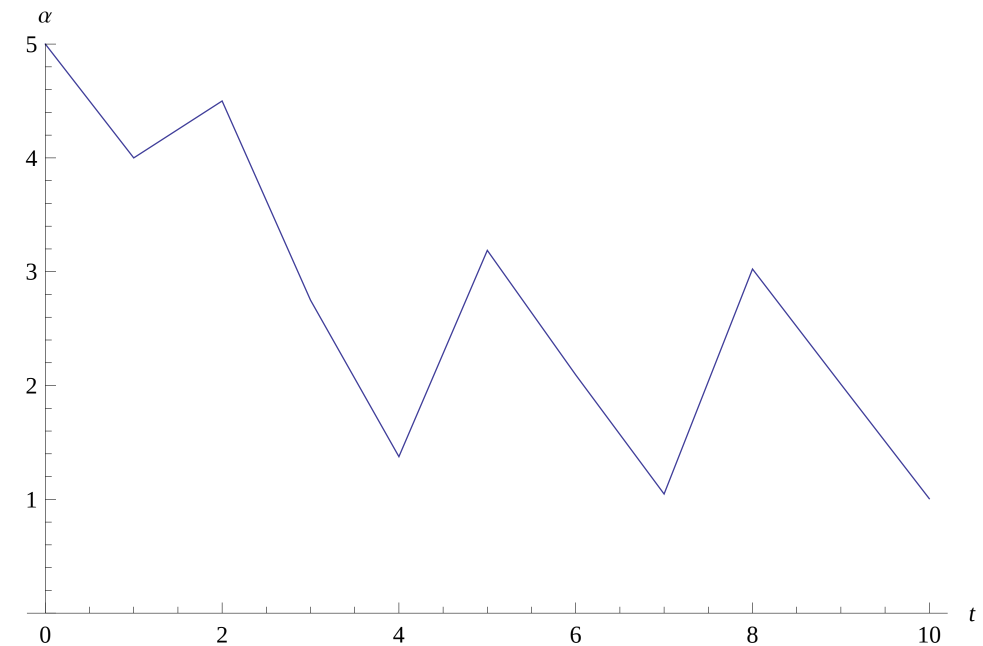


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D			

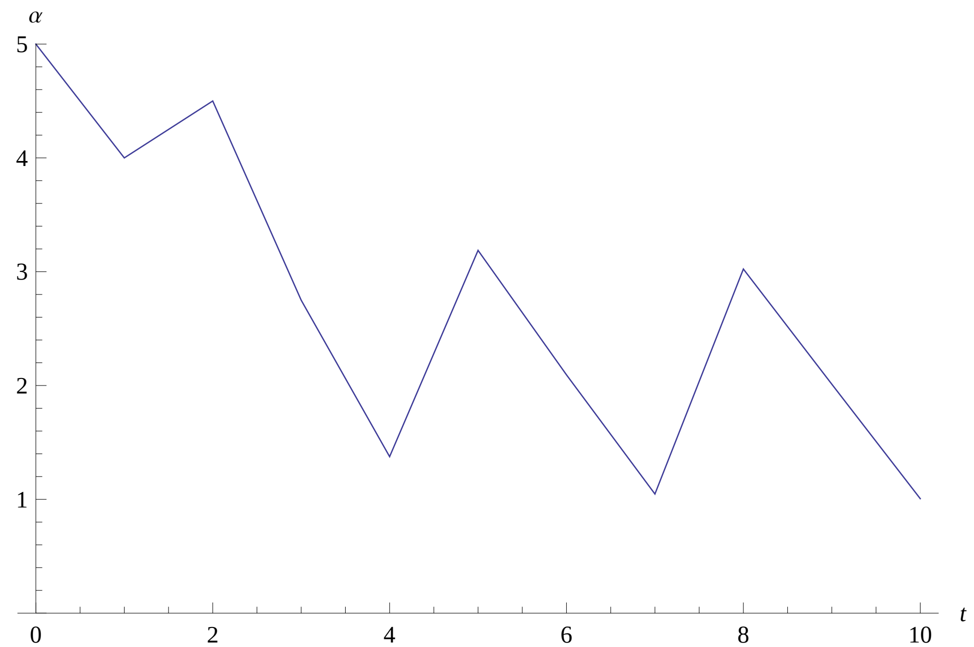


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**		

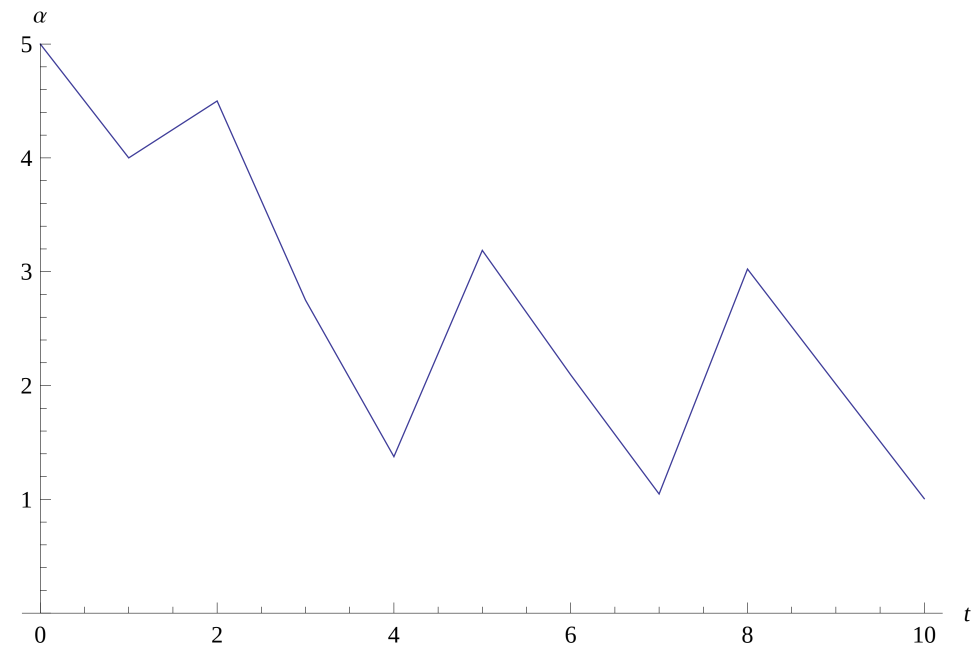


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	

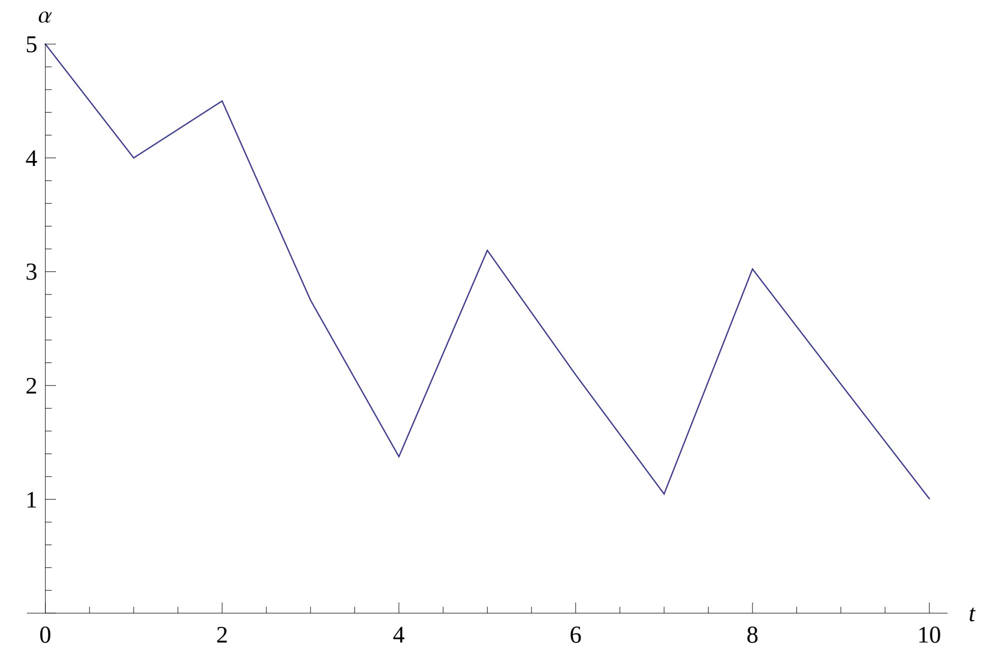


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75

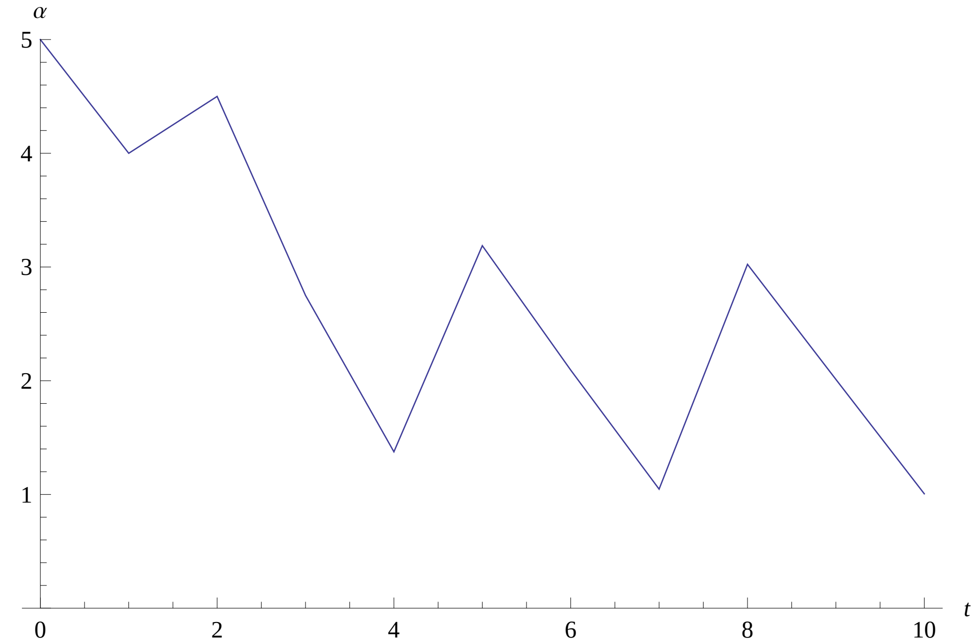


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4				

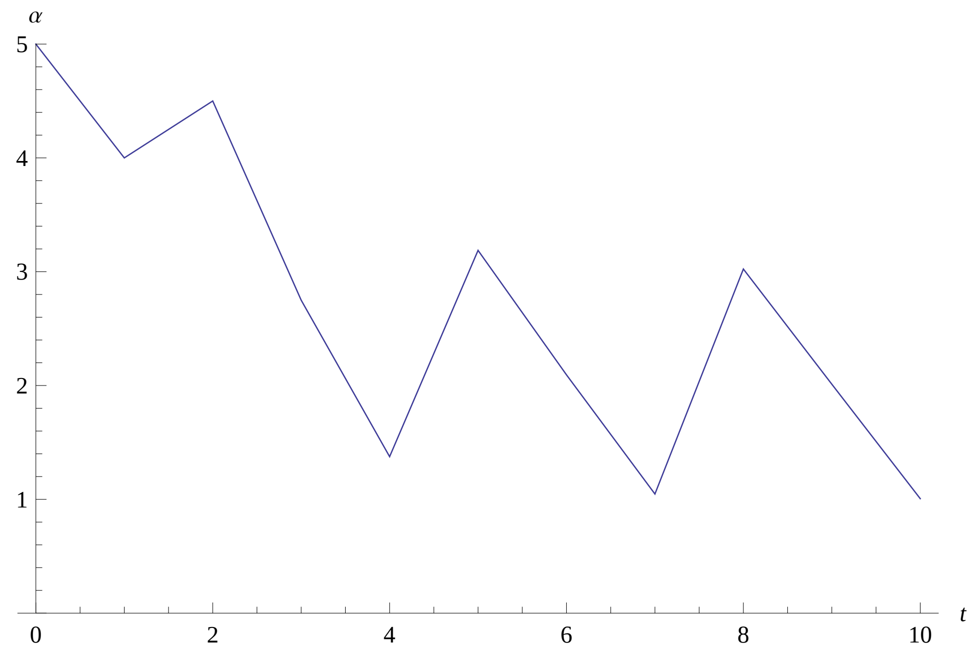


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C			

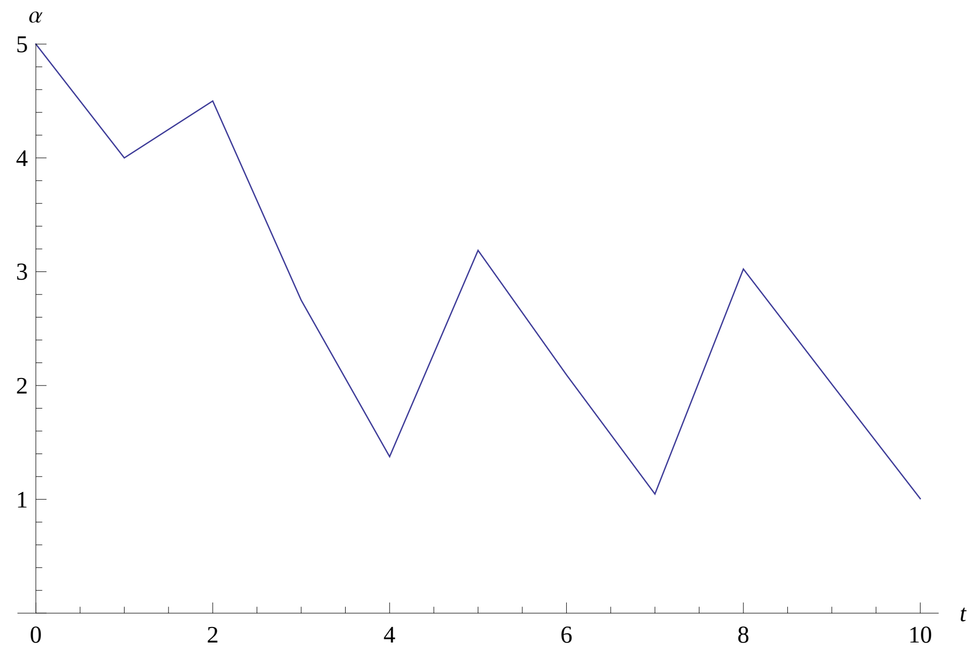


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**		

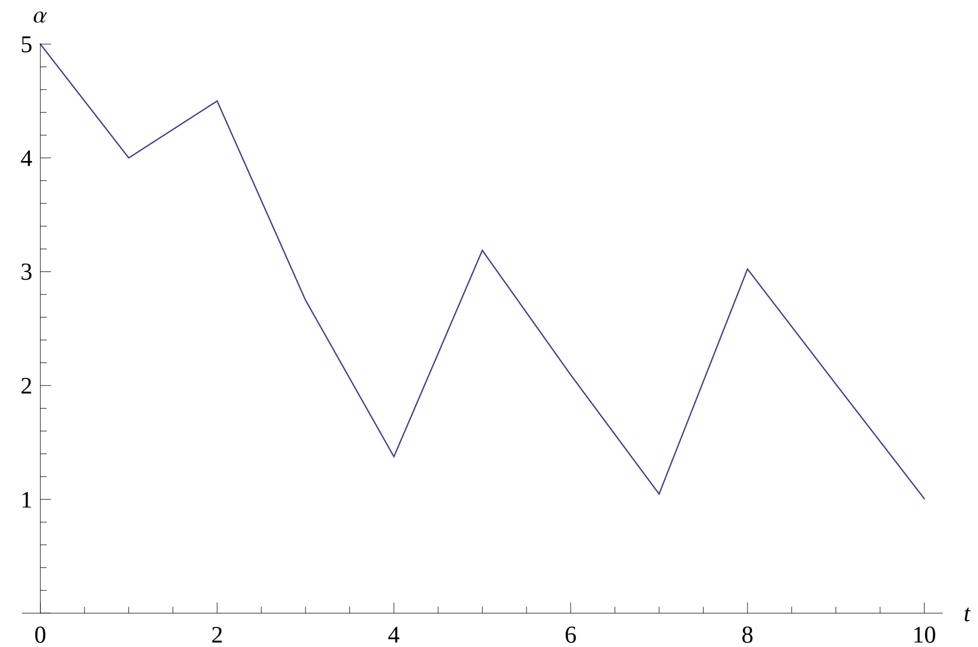


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	

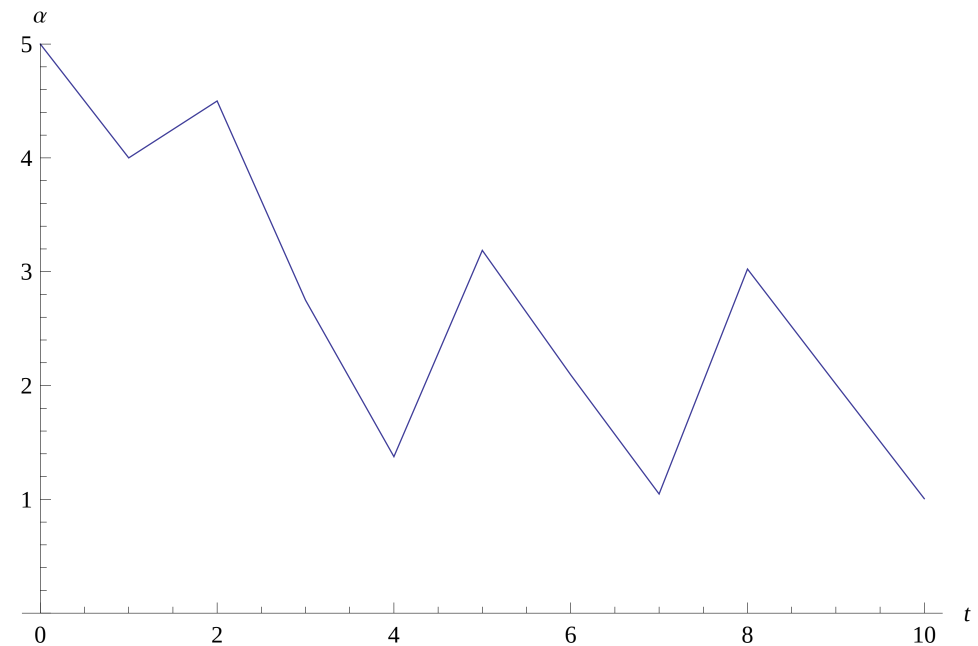


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375

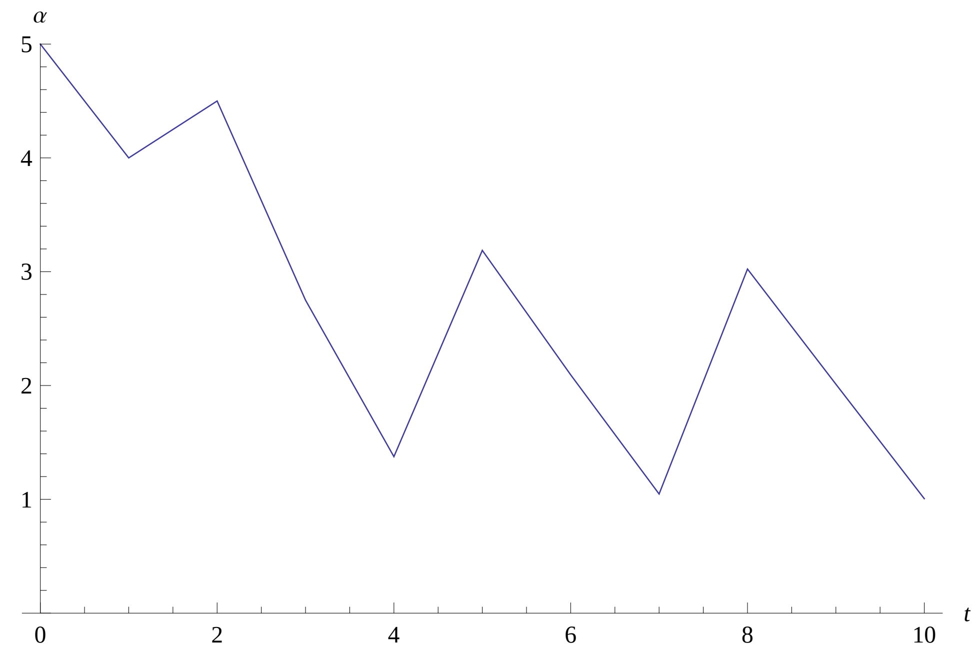


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5				

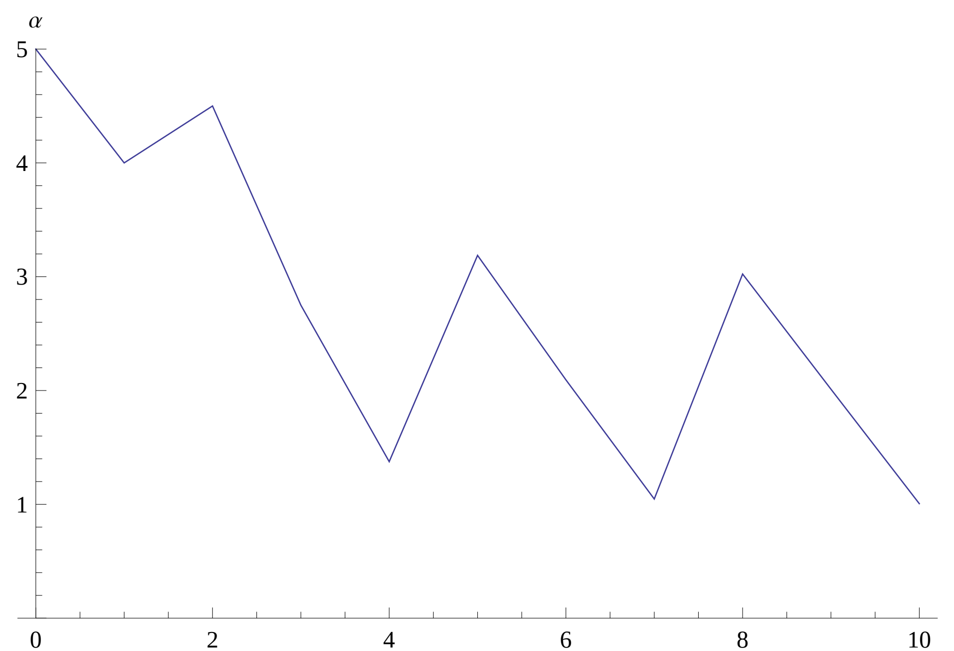


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D			

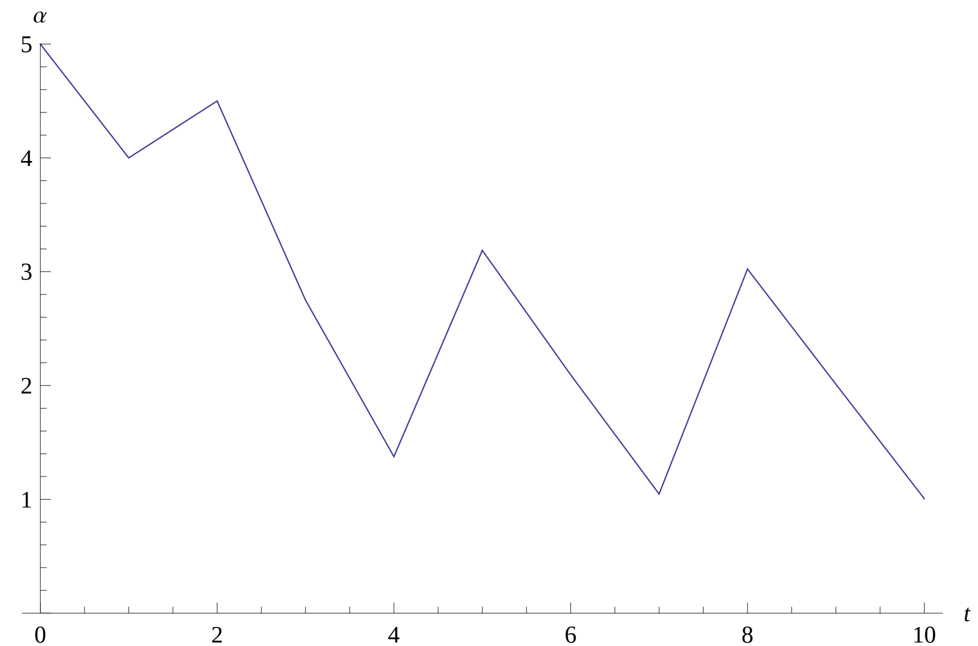


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D		

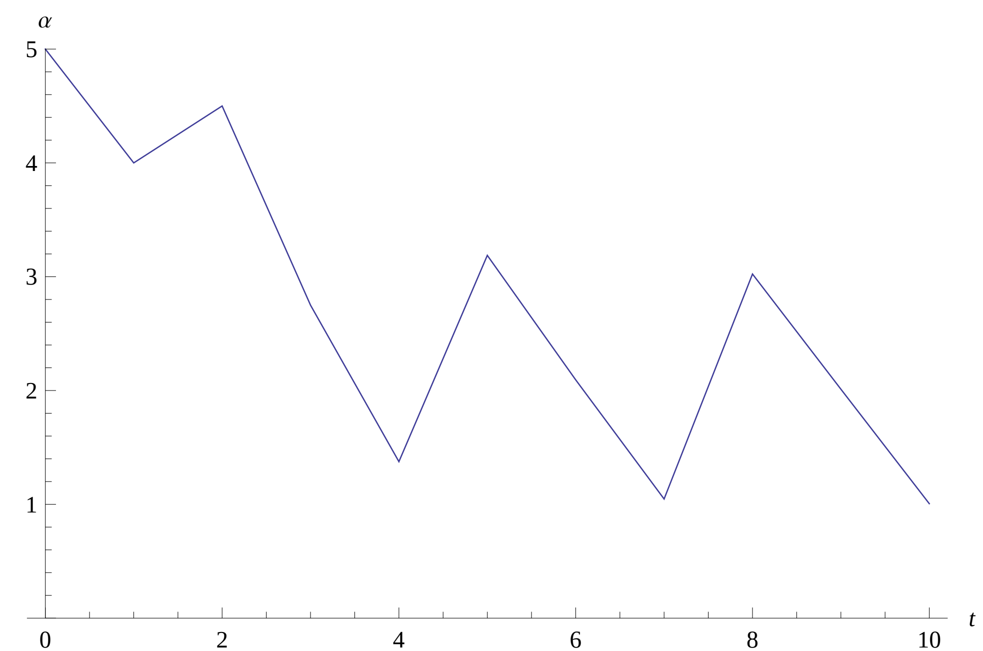


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	

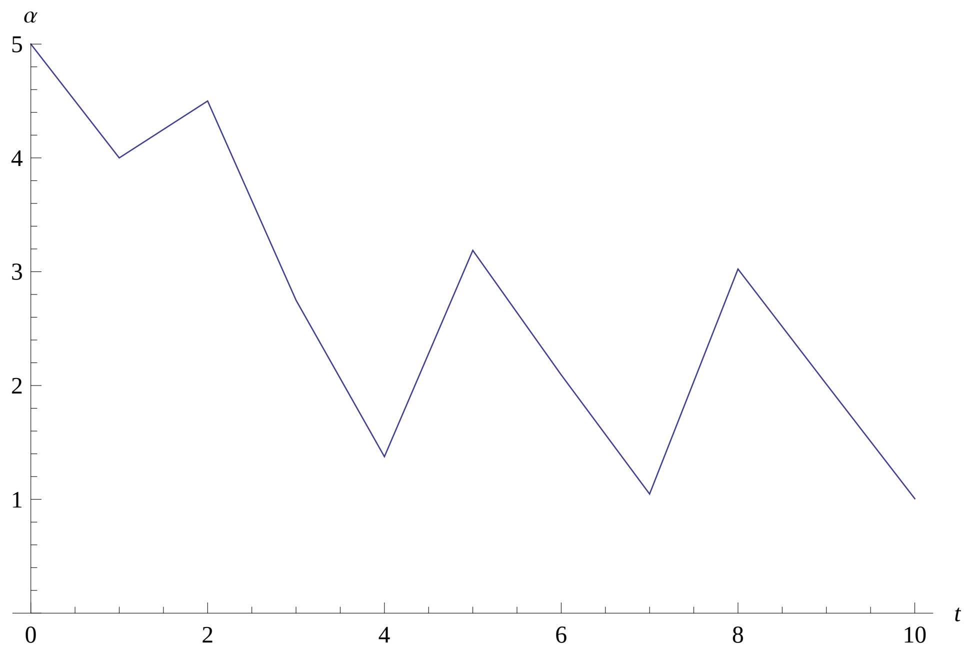


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875

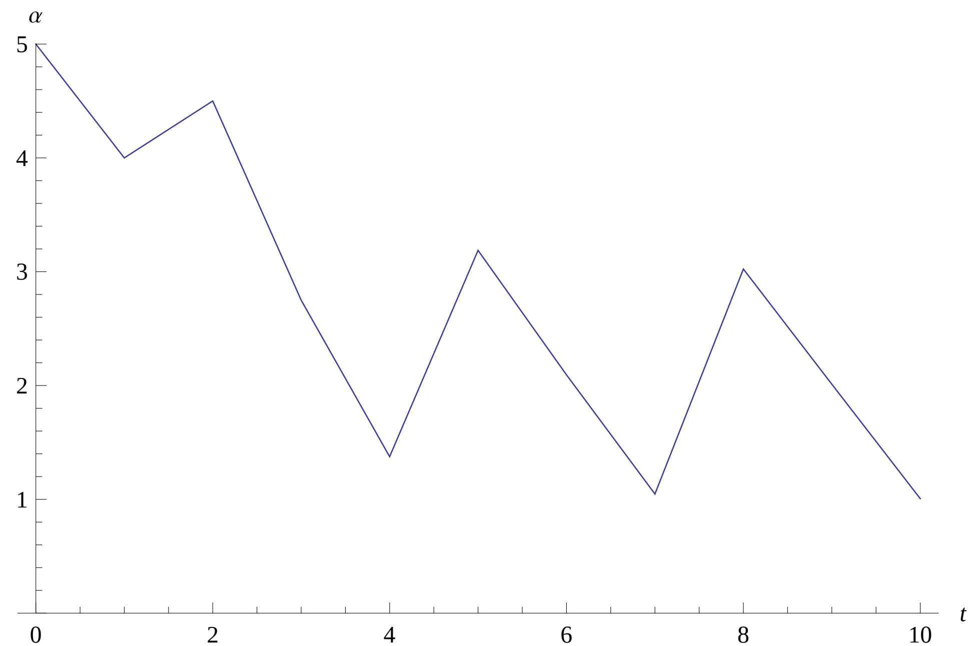


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6				

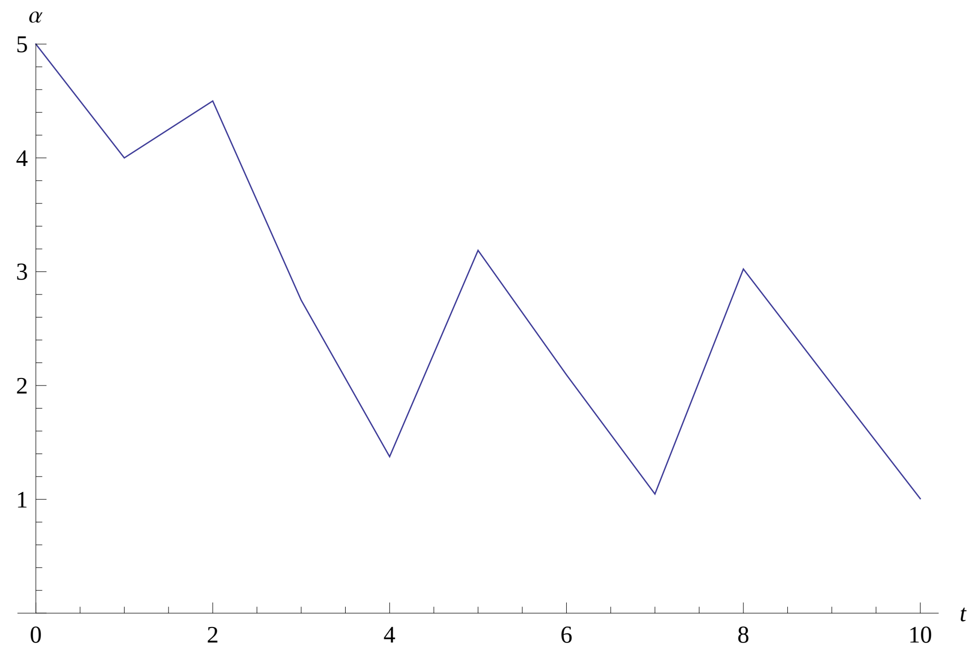


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D			

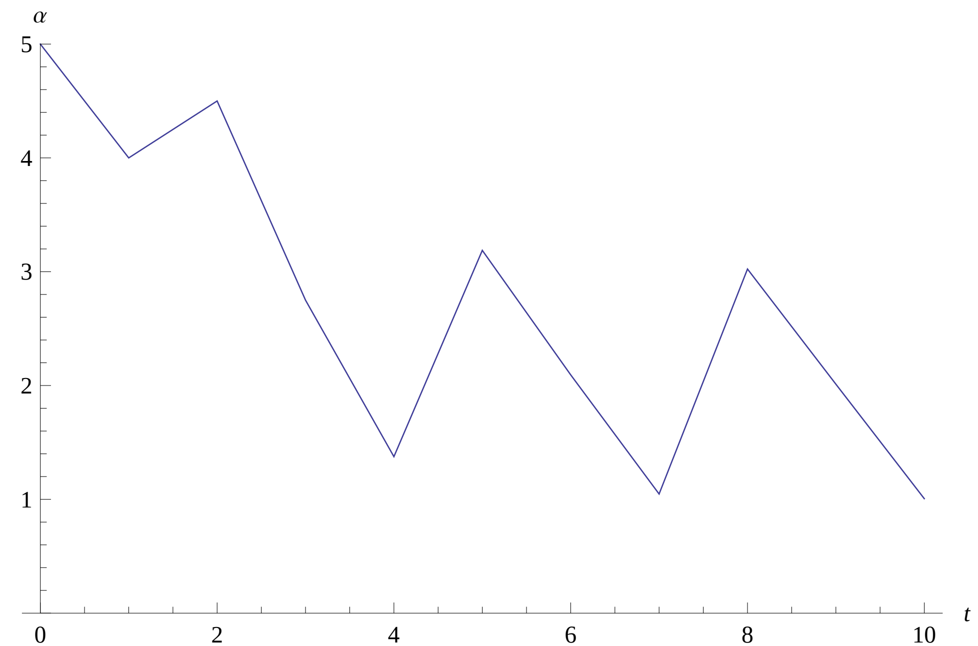


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**		

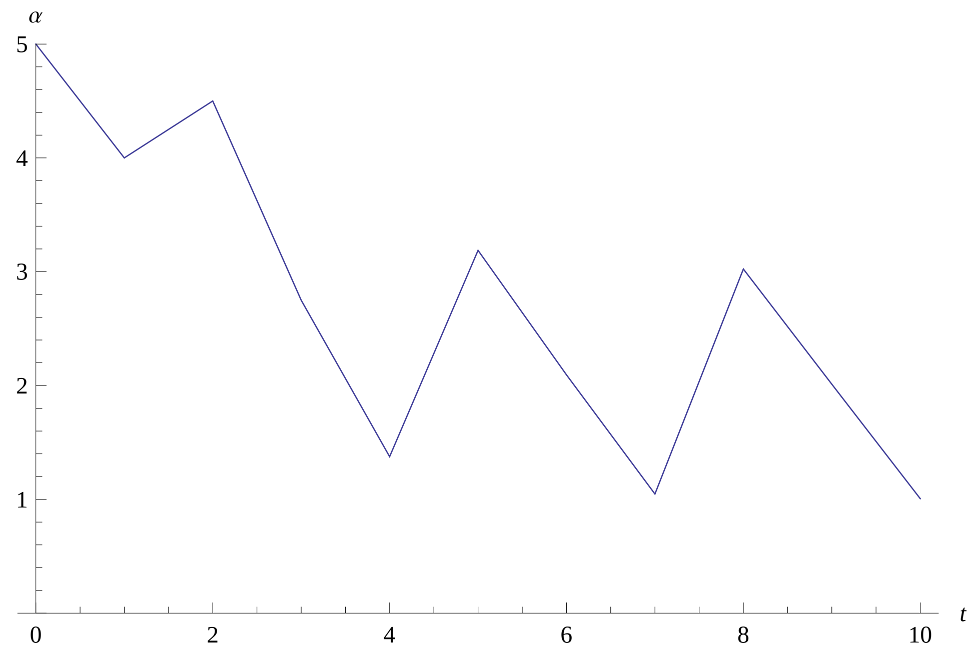


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	

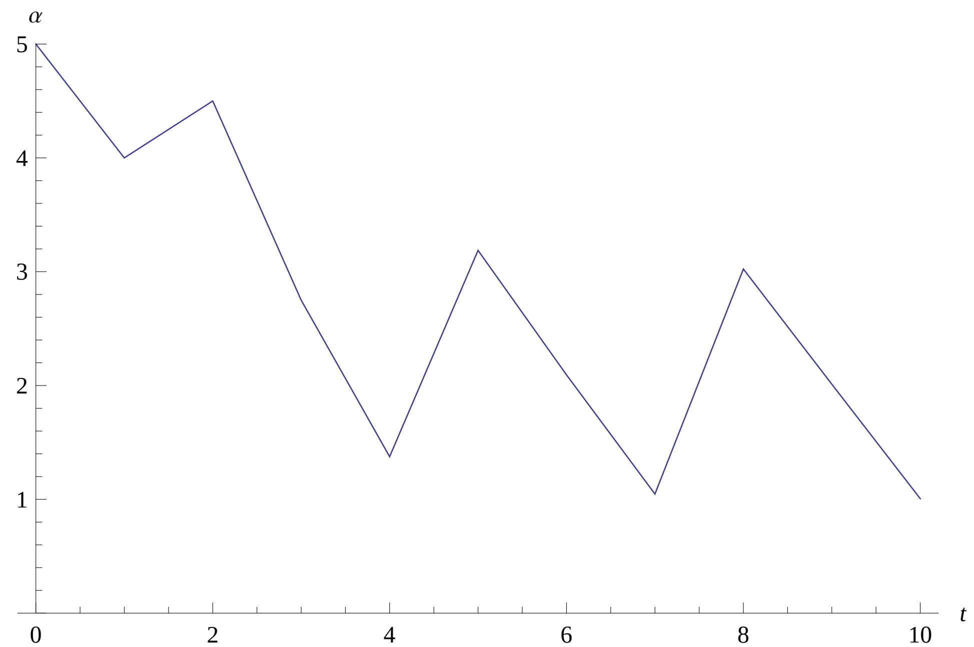


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375

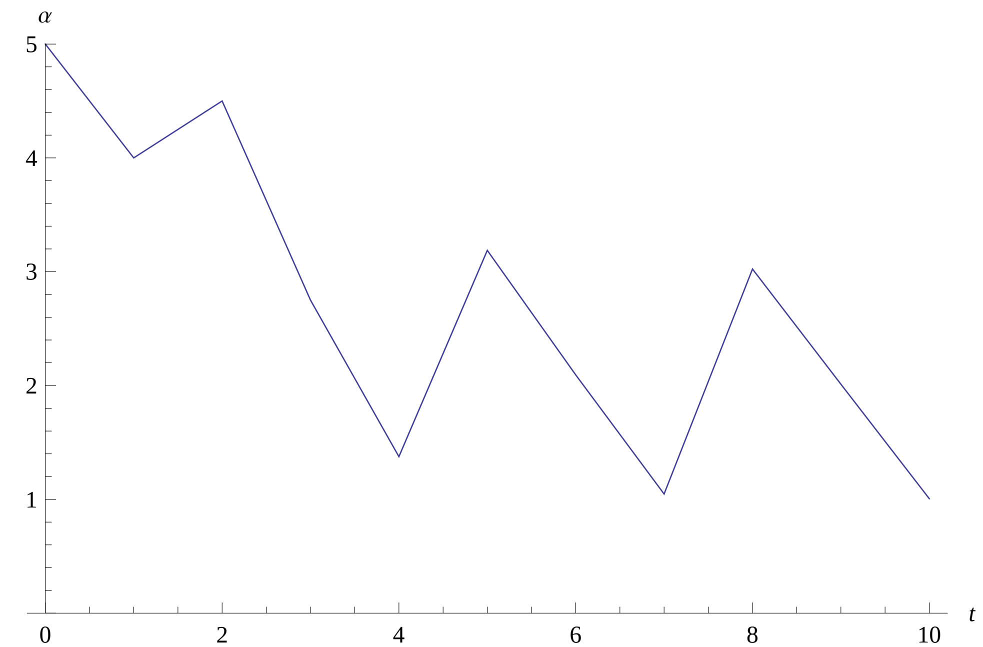


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7				

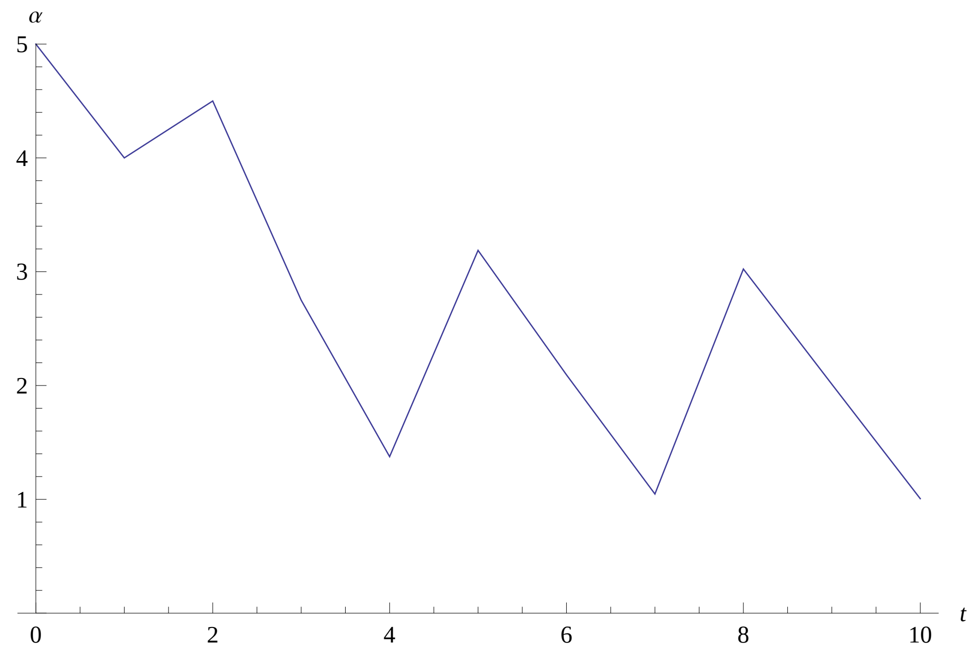


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C			

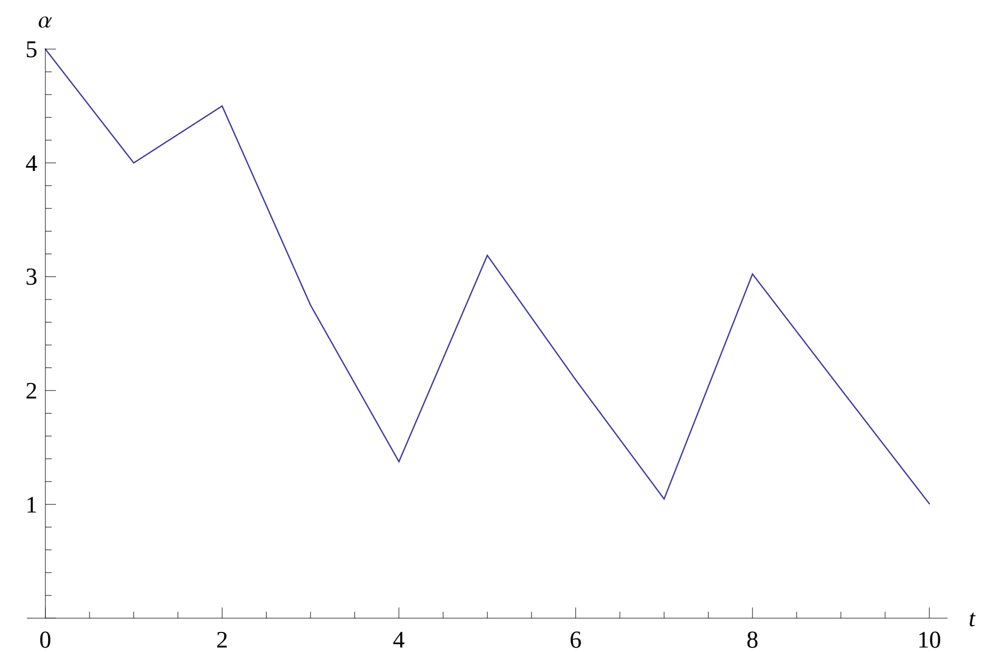


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**		

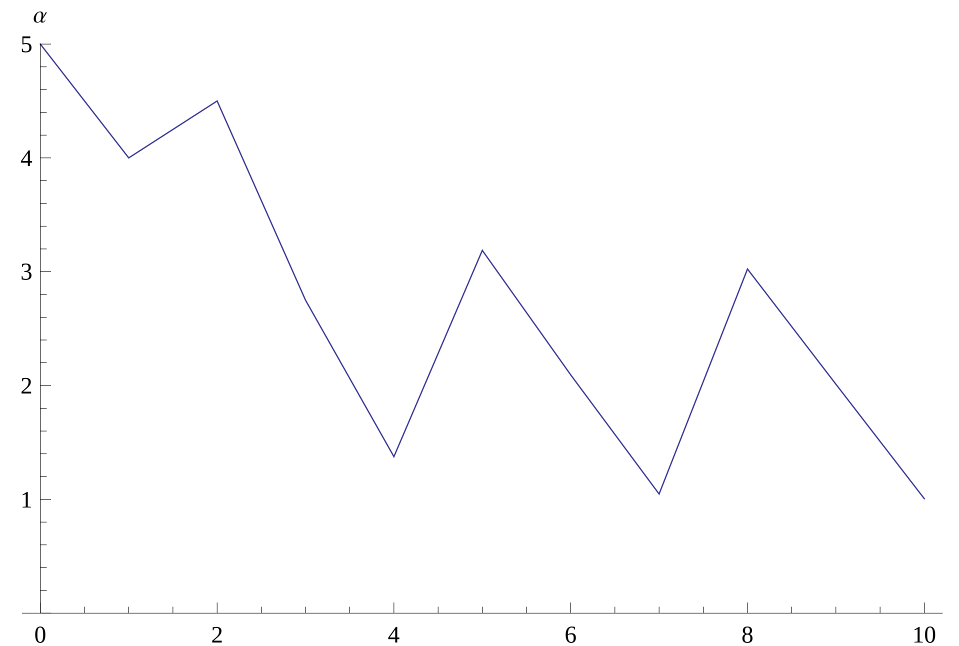


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	

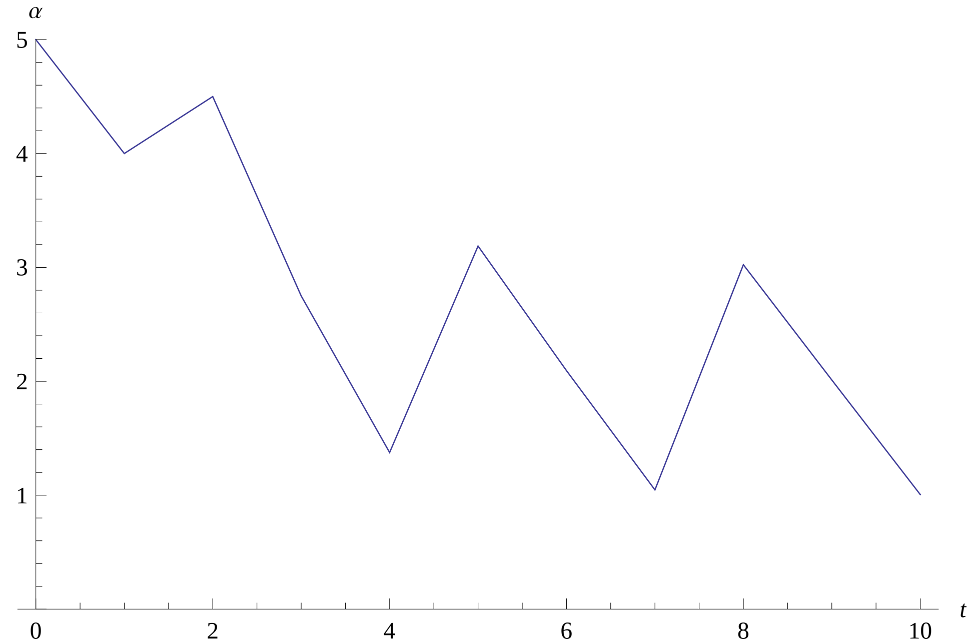


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875

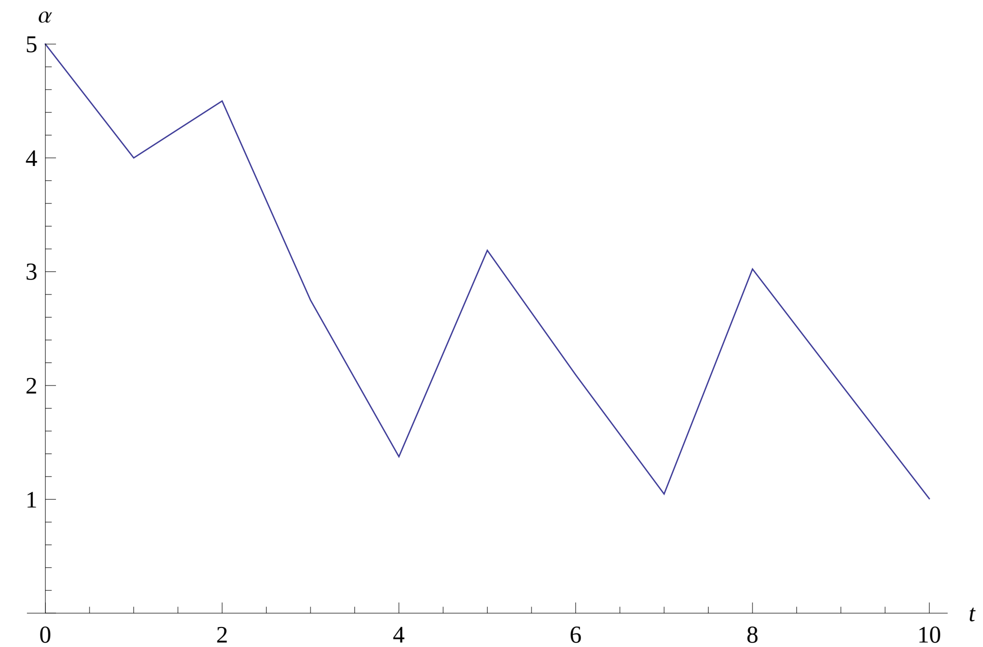


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8				

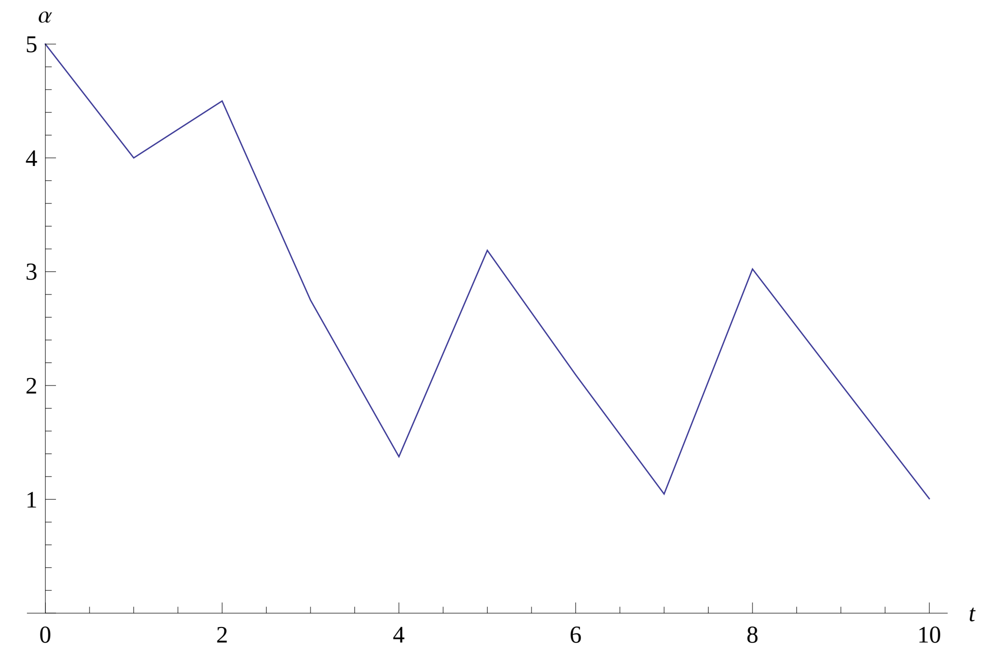


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D			

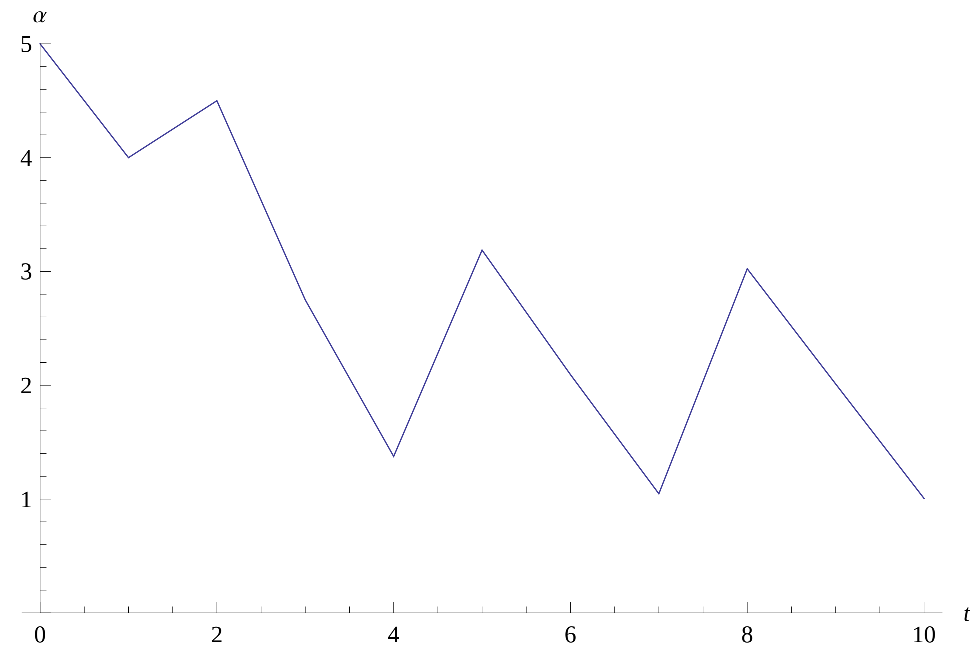


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D		

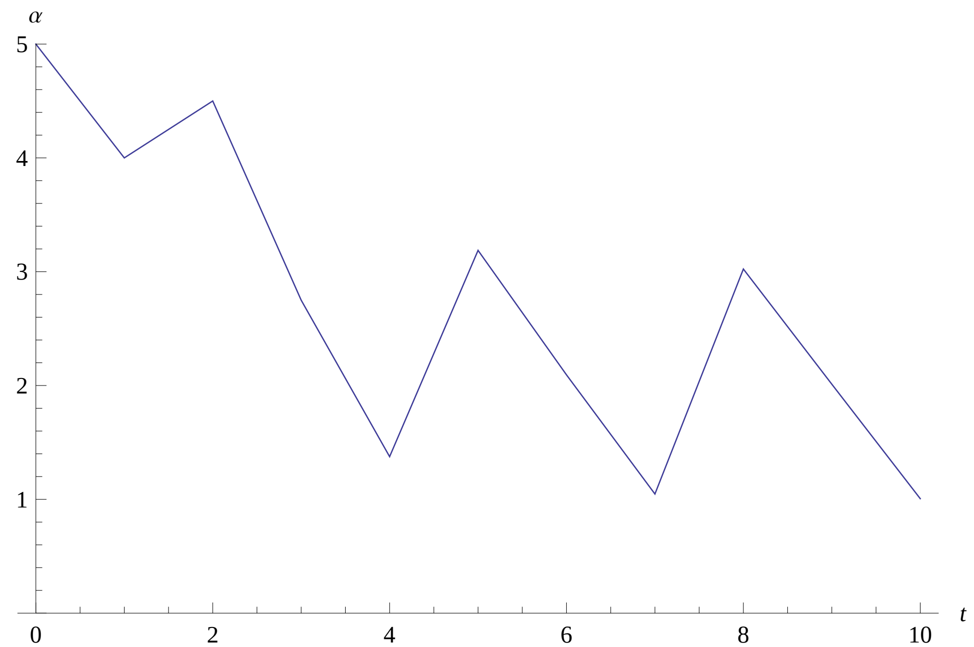


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	

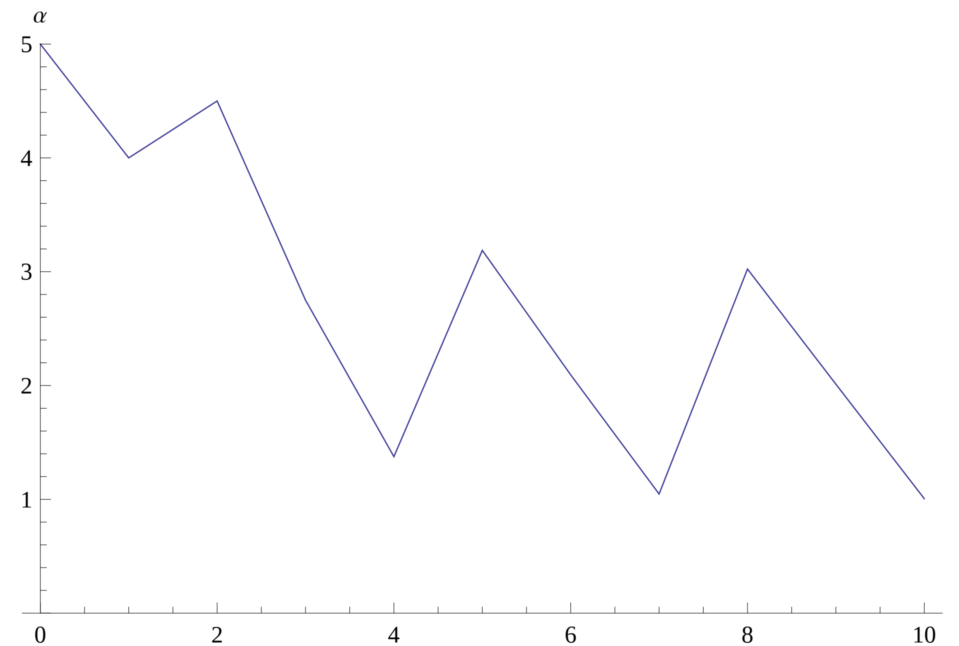


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375

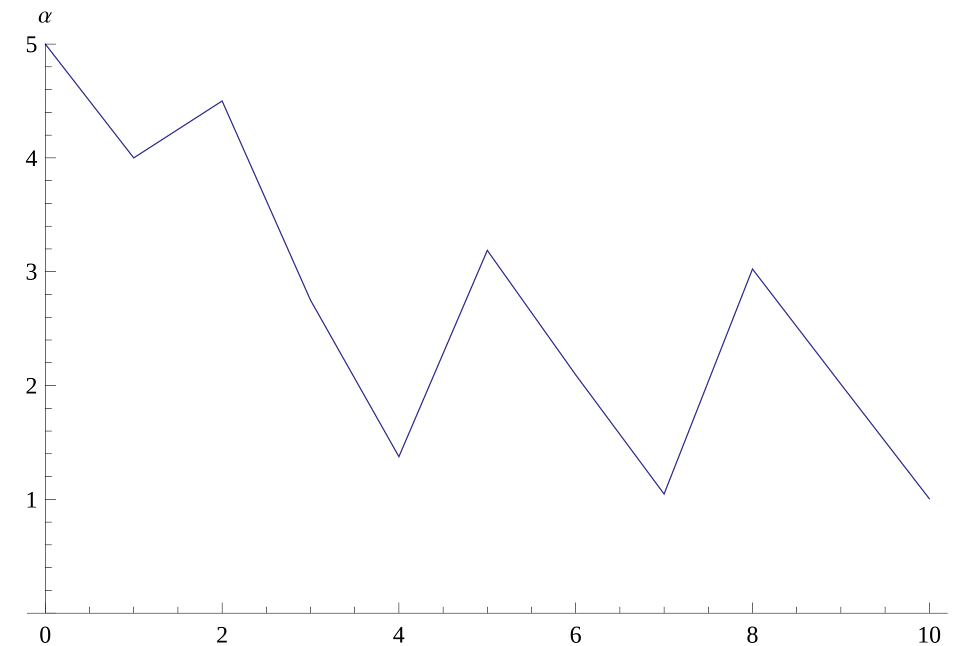


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9				

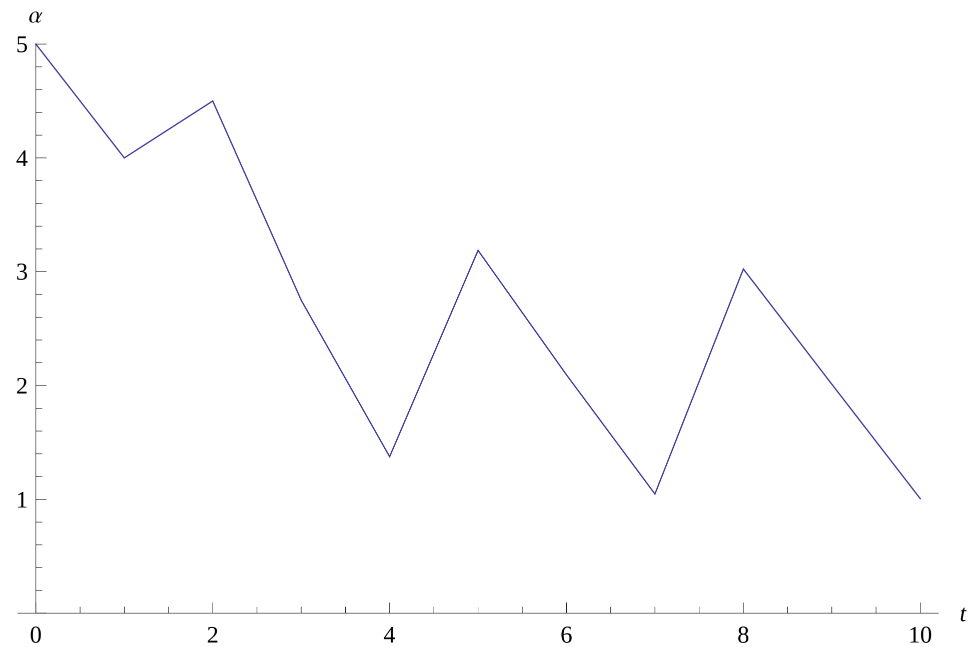


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9	D			

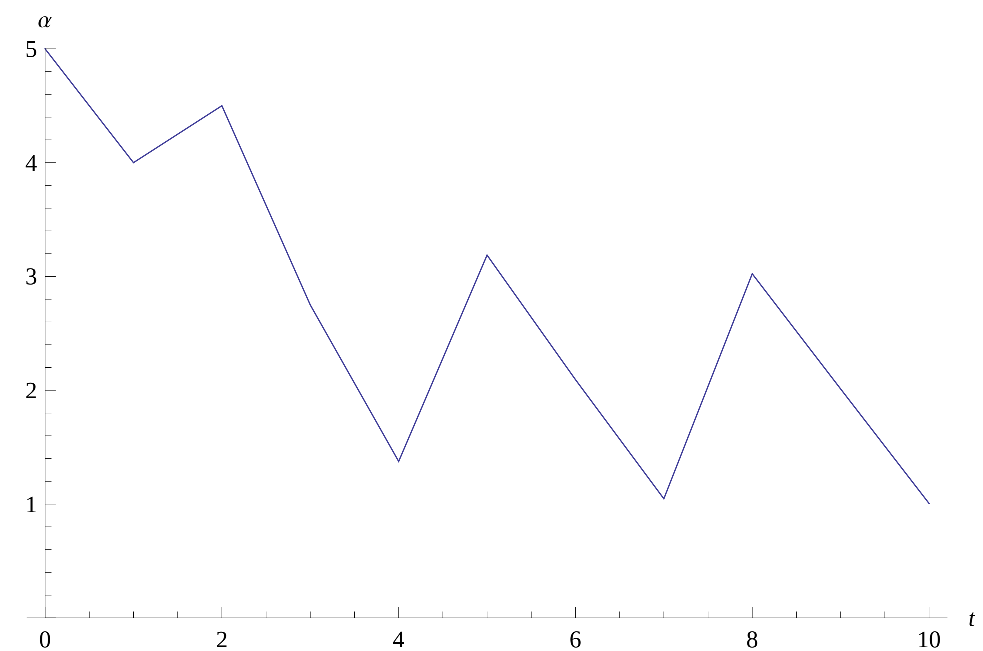


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9	D	**C**		

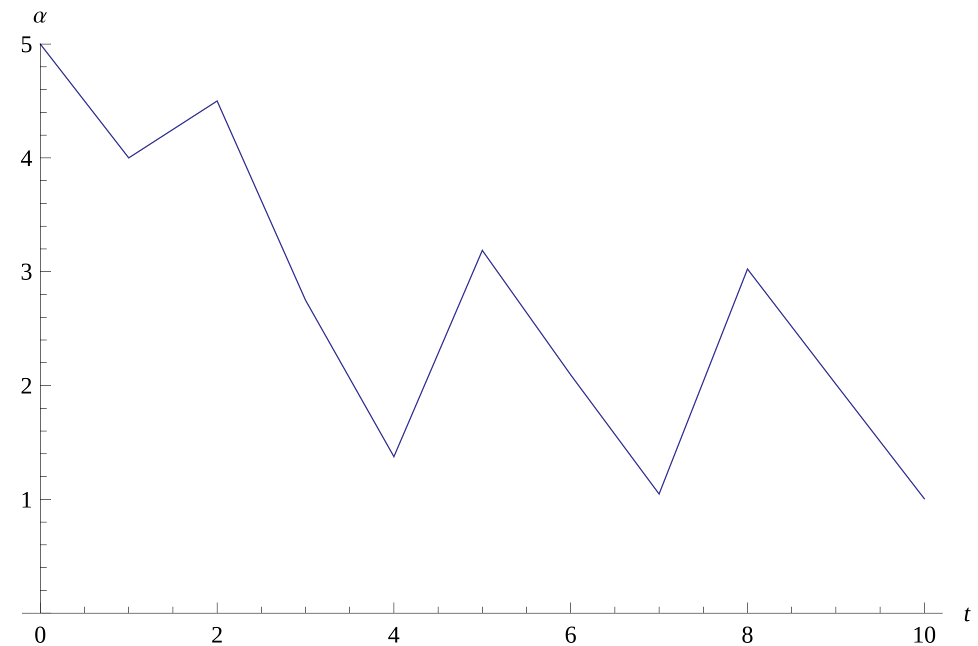


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9	D	**C**	0	

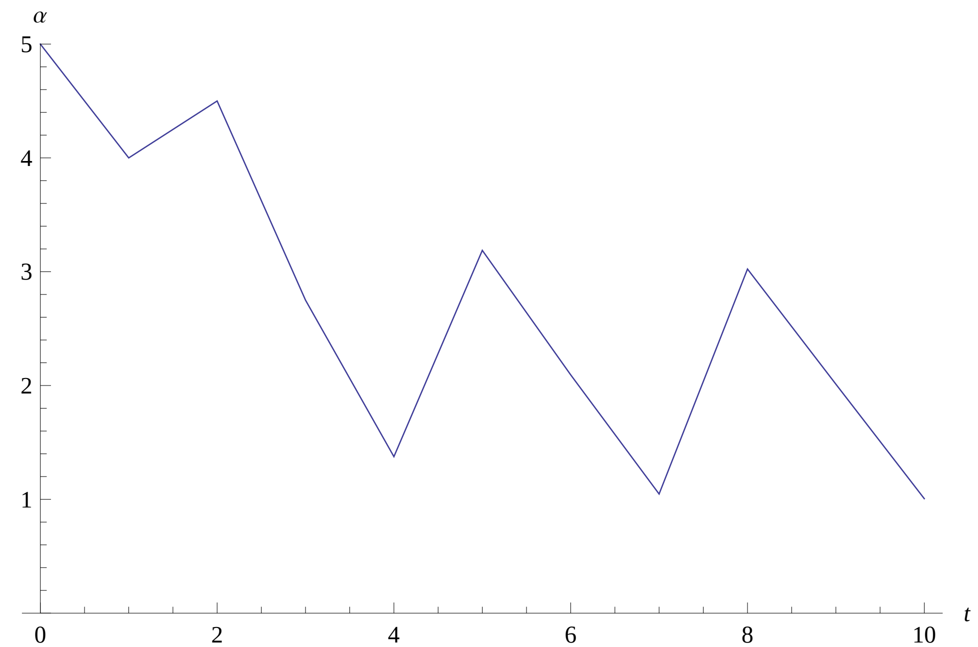


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9	D	**C**	0	2.01171875

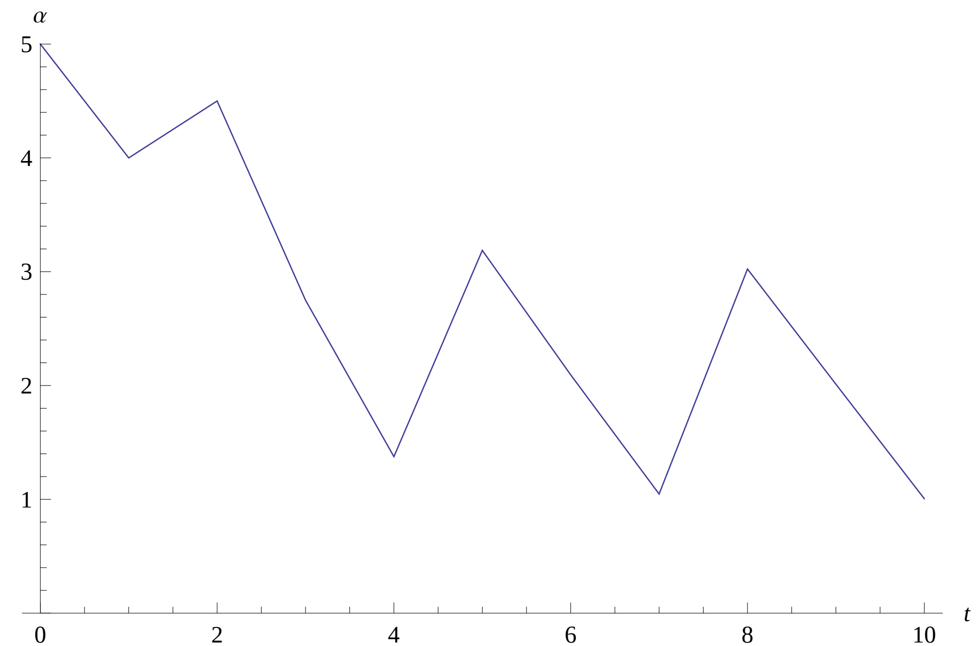


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9	D	**C**	0	2.01171875
10				

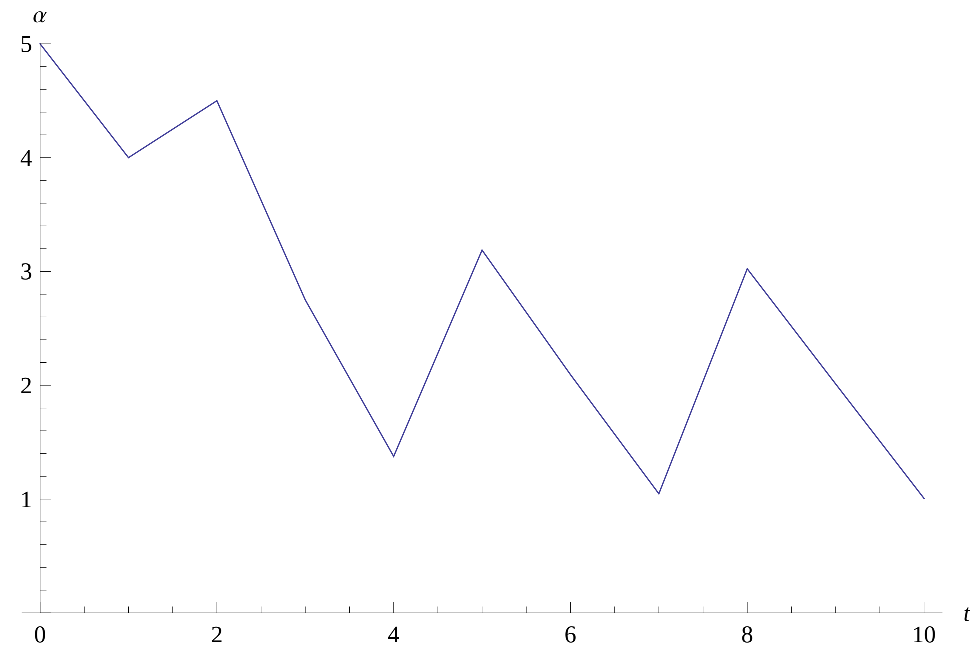


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9	D	**C**	0	2.01171875
10	C			

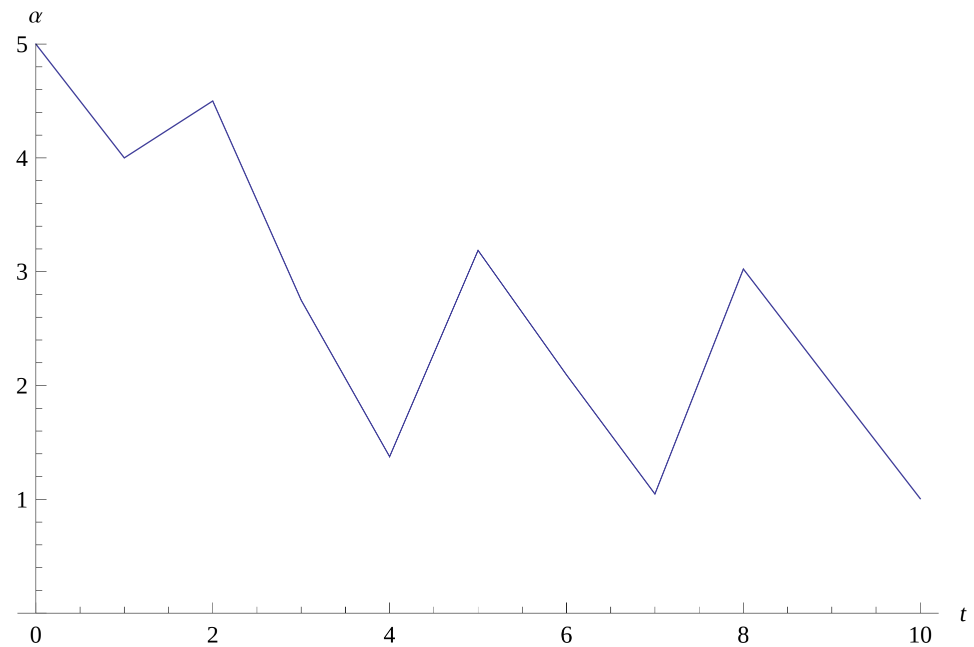


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9	D	**C**	0	2.01171875
10	C	D		

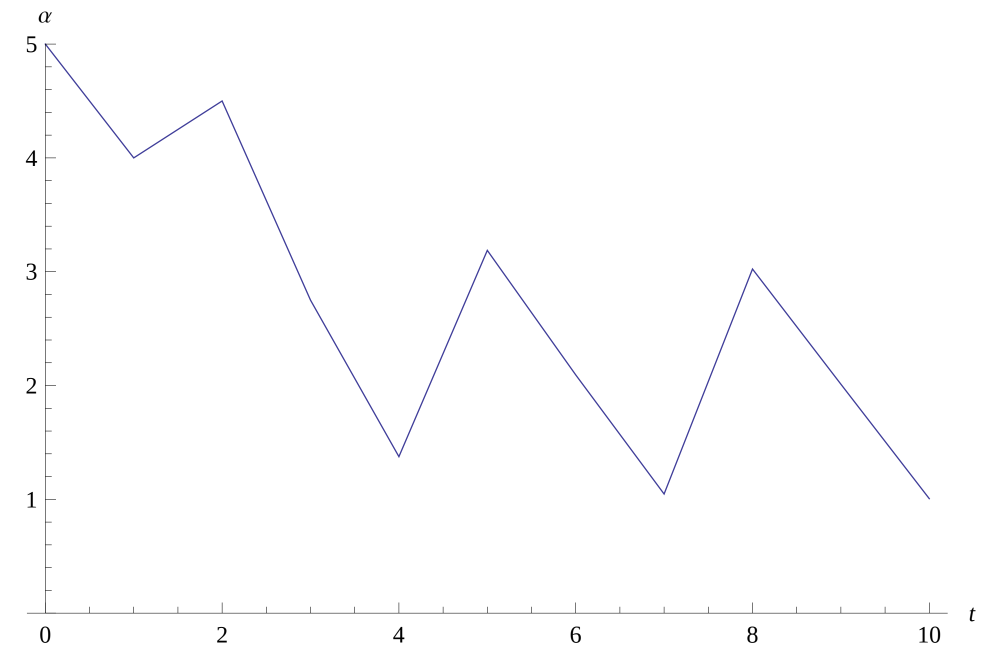


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9	D	**C**	0	2.01171875
10	C	D	5	

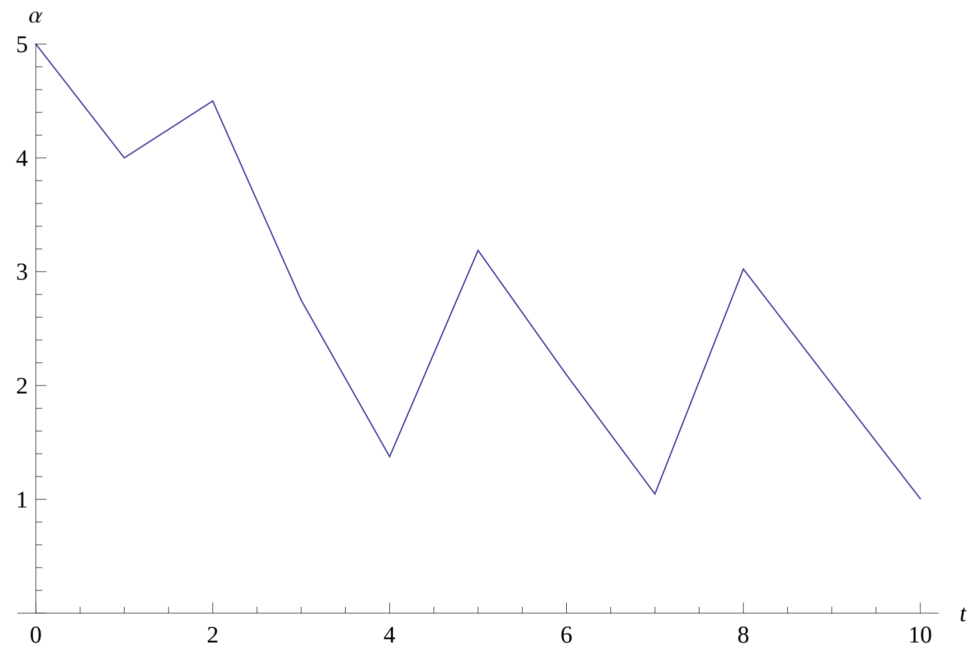


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9	D	**C**	0	2.01171875
10	C	D	5	1.005859375

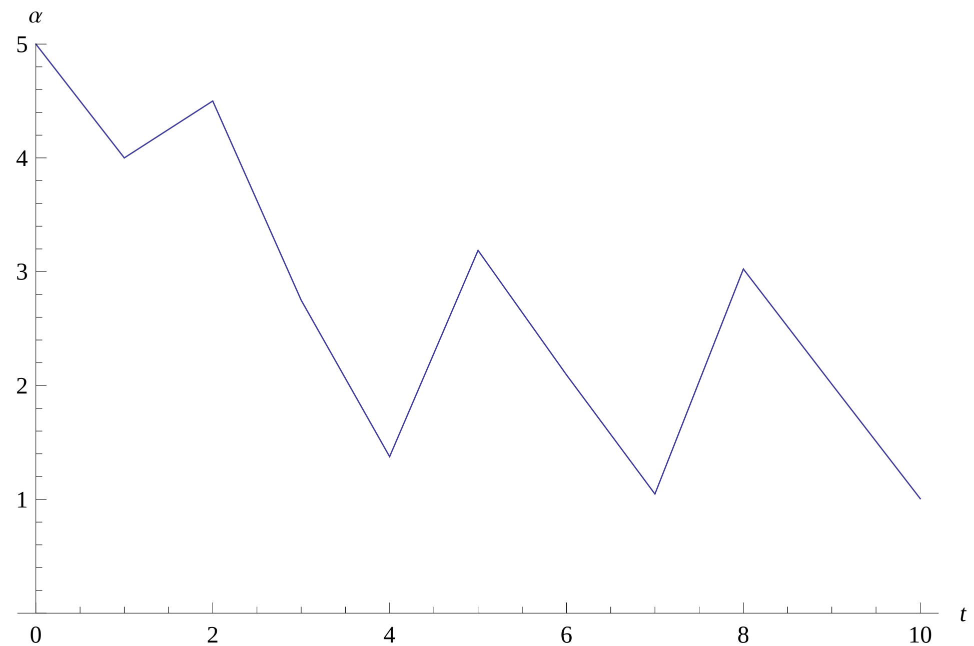


Progress of aspirations.

Example of satisficing play

Game: prisoner's dilemma. Strategy player 1: tit-for-tat. Strategy player 2: satisficing with initial state $(A_0, \alpha_0) = (C, 5)$. Persistence rate: $\lambda = 0.5$.

t	TFT	A_t	π_t	α_t
0	C	C	3	5
1	C	**D**	5	4
2	D	D	1	4.5
3	D	**C**	0	2.75
4	C	**D**	5	1.375
5	D	D	1	3.1875
6	D	**C**	0	2.09375
7	C	**D**	5	1.046875
8	D	D	1	3.0234375
9	D	**C**	0	2.01171875
10	C	D	5	1.005859375
	⋮	⋮	⋮	⋮

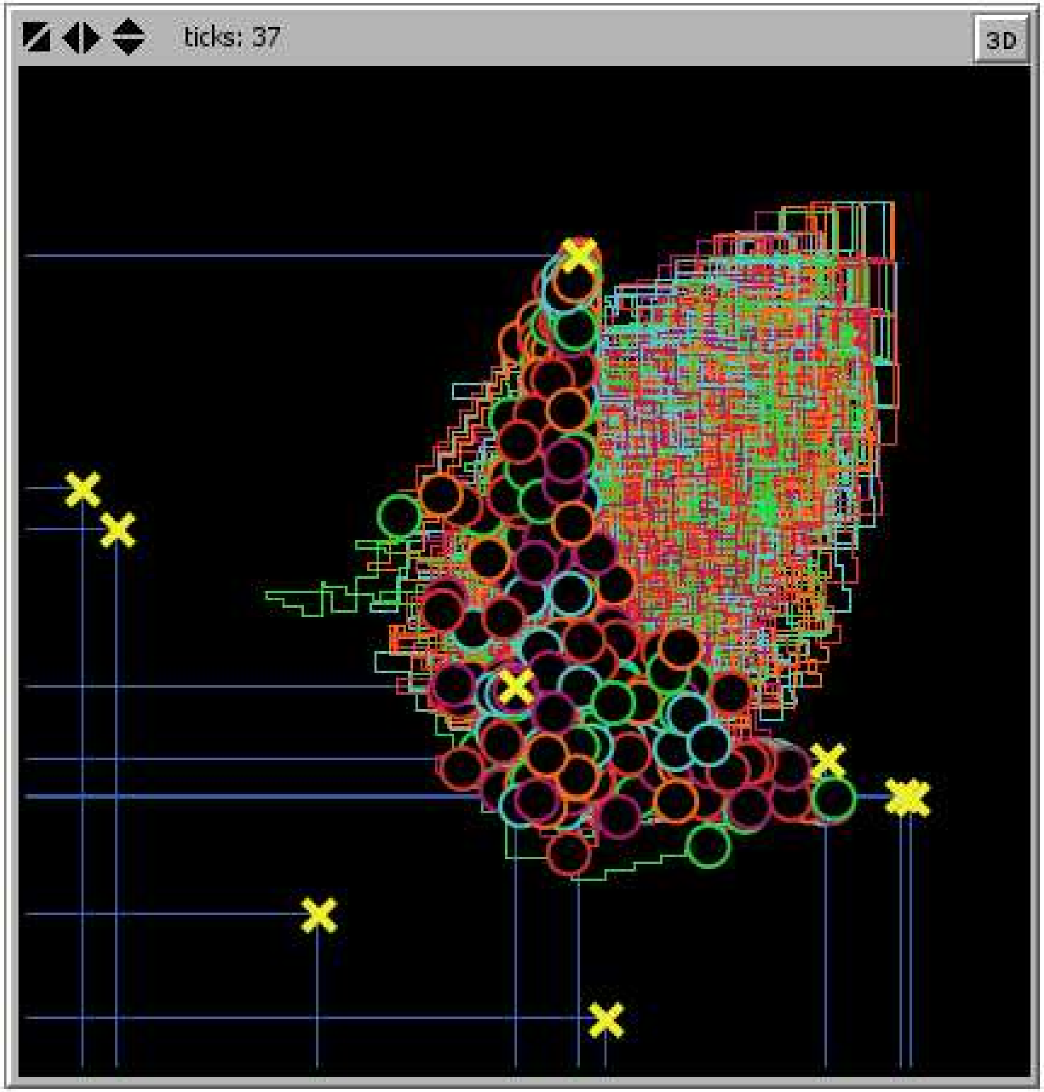


Progress of aspirations.

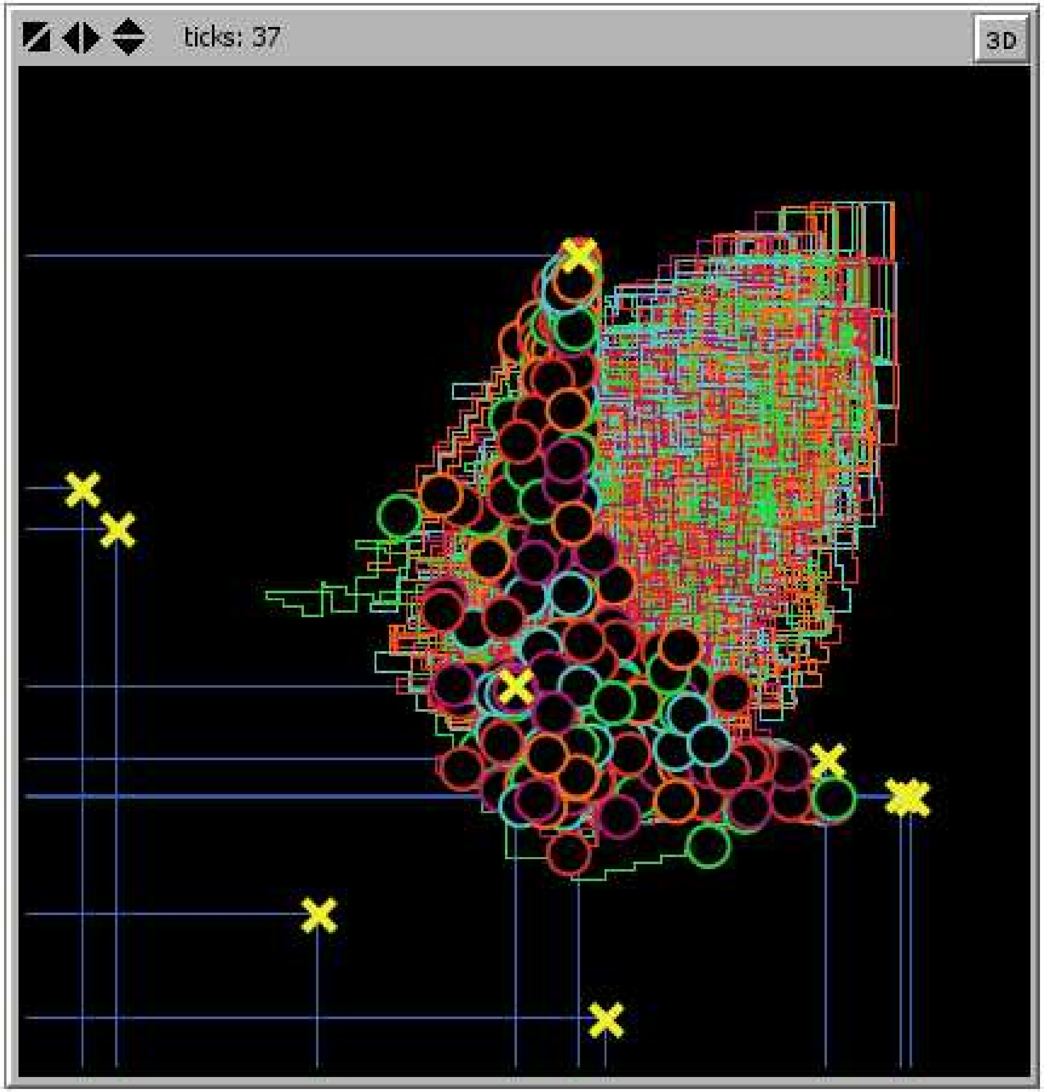
Demo:

Satisficing play in general 2-player 3x3 matrix games

Approach

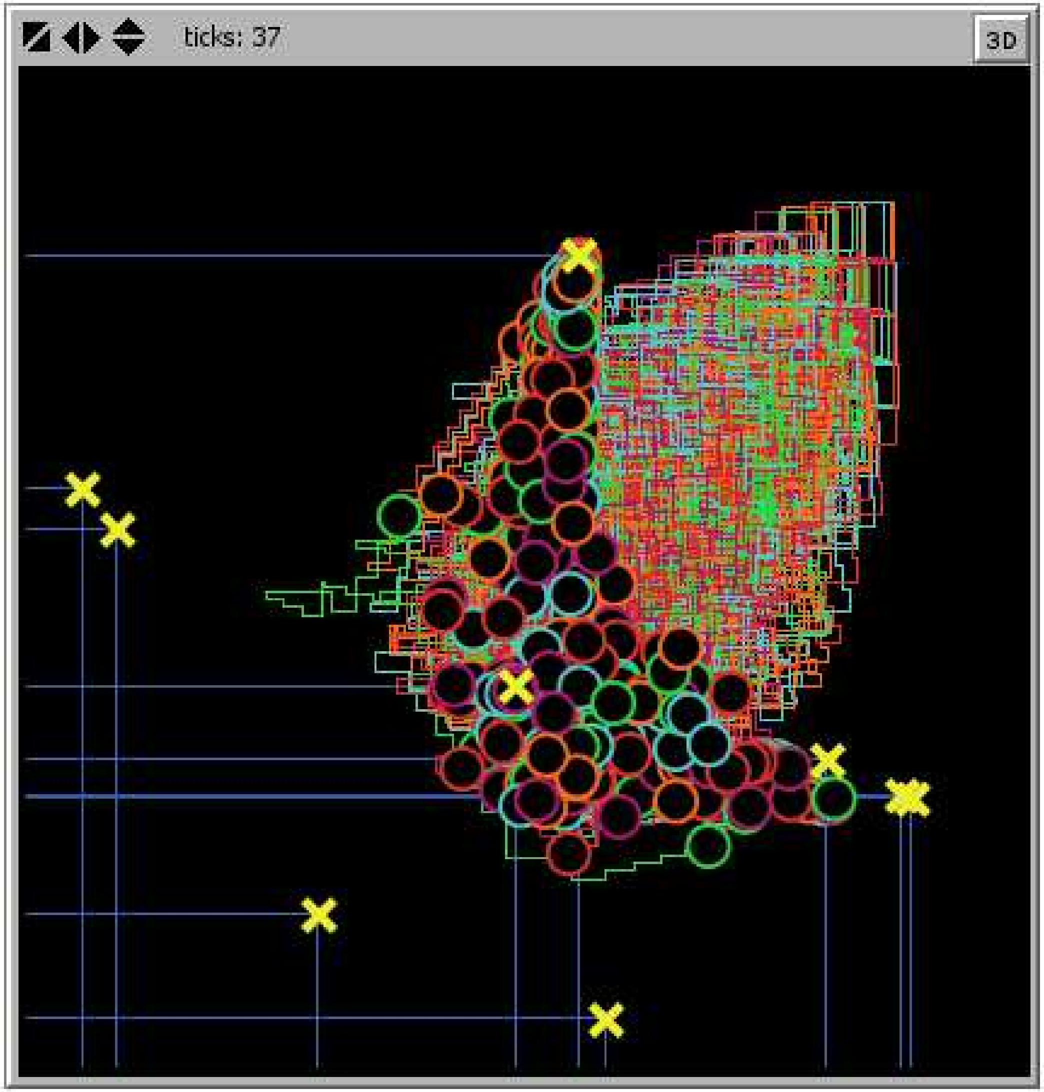


Approach



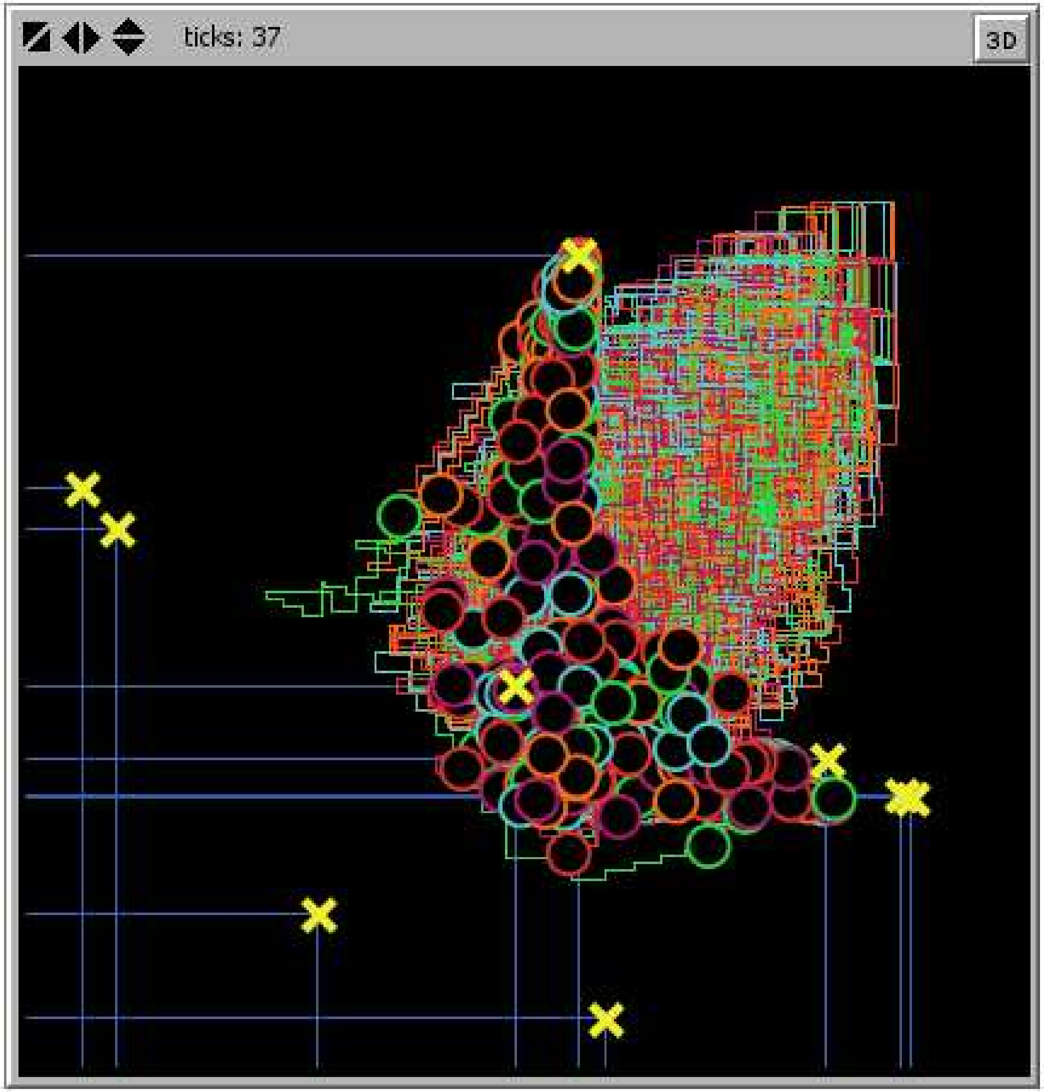
- Take a 2-player 3×3 game in normal form.

Approach



- Take a 2-player 3×3 game in normal form.
- Plot all 9 pure payoff profiles in 2D.

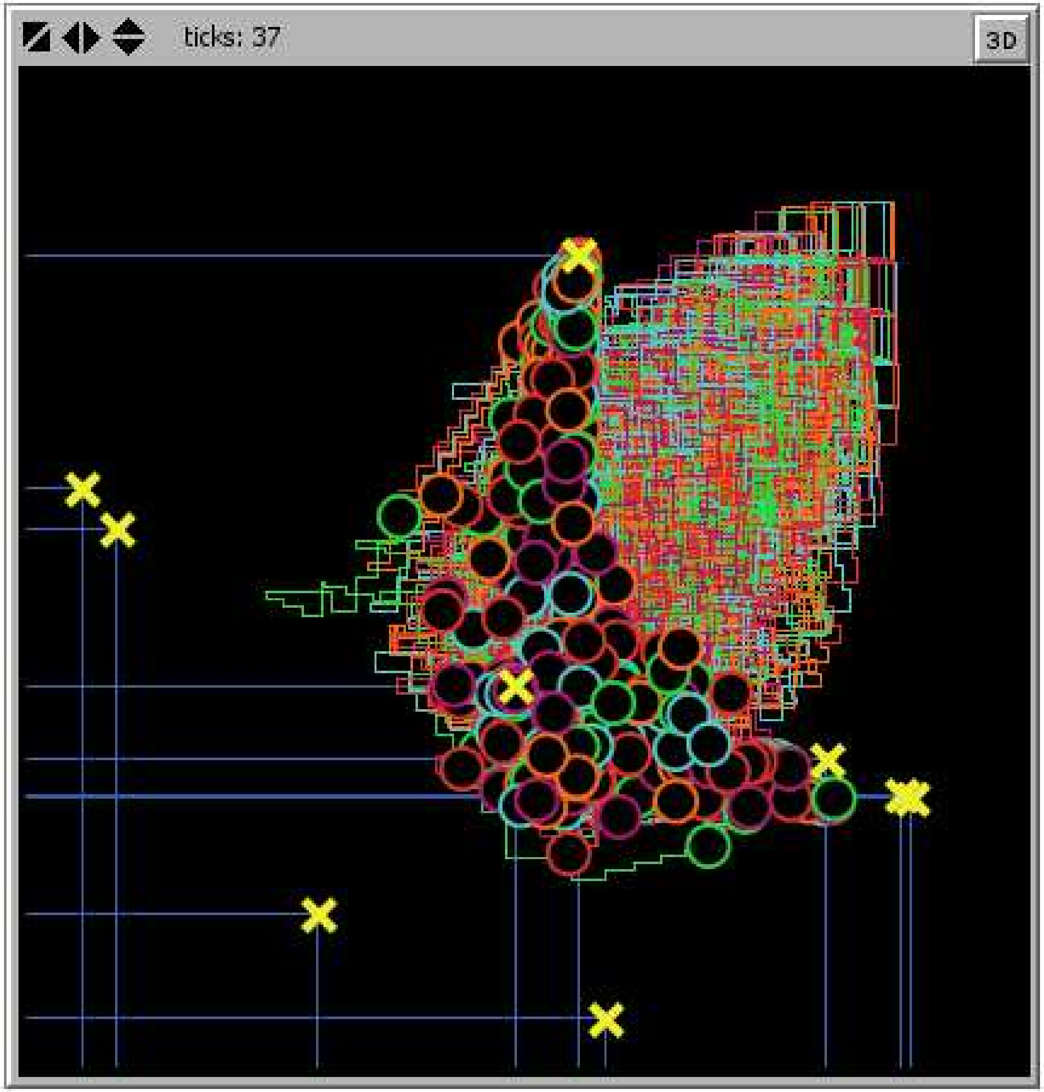
Approach



- Take a 2-player 3×3 game in normal form.
- Plot all 9 pure payoff profiles in 2D.
- Initialize, say, 100 profiles. One profile looks like:

$$(\ (A_t, \alpha_t) \ , \ (B_t, \beta_t) \ ).$$

Approach

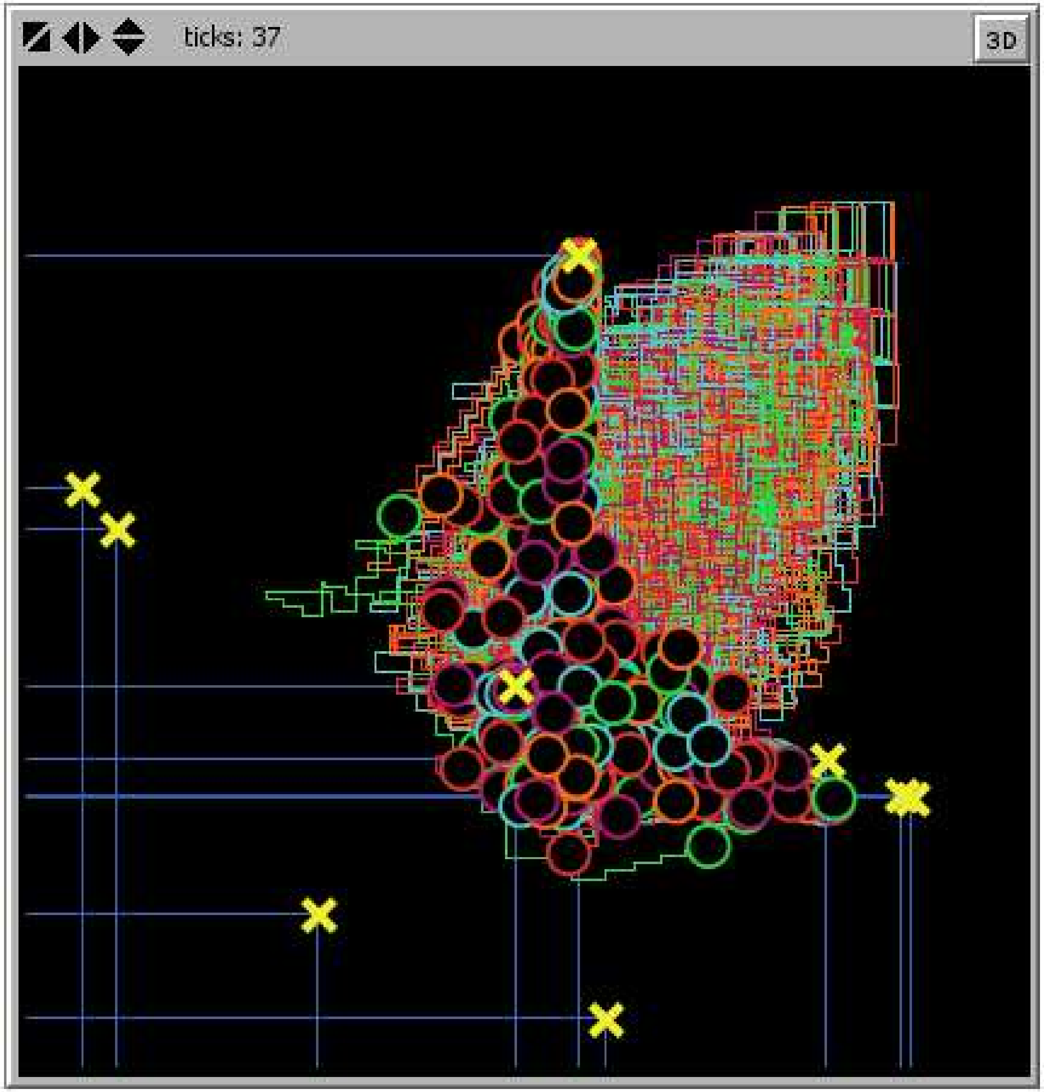


- Take a 2-player 3×3 game in normal form.
- Plot all 9 pure payoff profiles in 2D.
- Initialize, say, 100 profiles. One profile looks like:

$$(\ (A_t, \alpha_t)\ ,\ (B_t, \beta_t)\ ).$$

Plot the corresponding 100 aspiration profiles (α_t, β_t) in the same canvas.

Approach

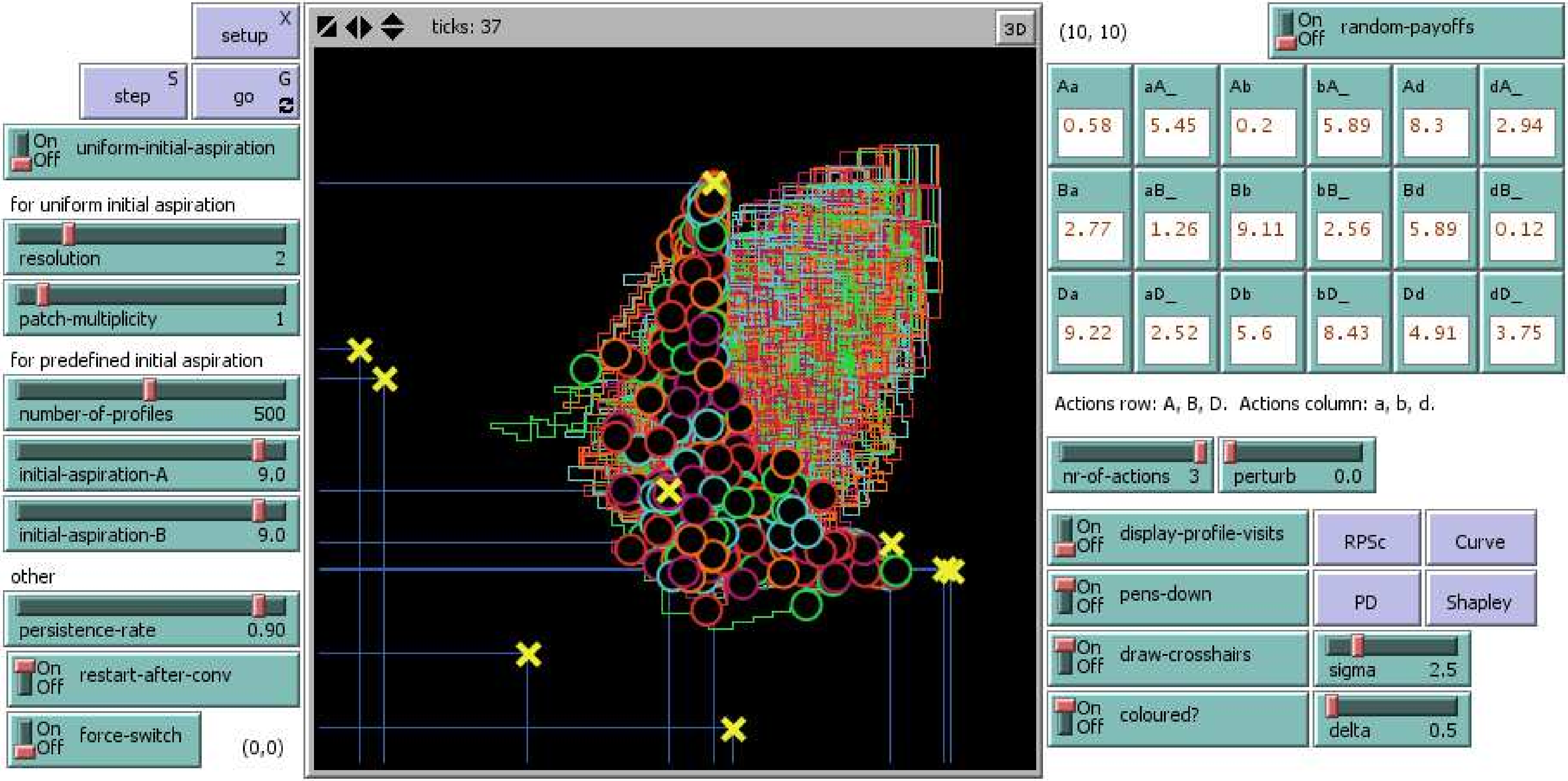


- Take a 2-player 3×3 game in normal form.
- Plot all 9 pure payoff profiles in 2D.
- Initialize, say, 100 profiles. One profile looks like:

$$(\ (A_t, \alpha_t) \ , \ (B_t, \beta_t) \ ).$$

  Plot the corresponding 100 aspiration profiles (α_t, β_t) in the same canvas.
- Execute satisficing play for all player profiles simultaneously.

Satisficing play in a 2-player matrix game



Satisficing play in a generalised prisoner's dilemma with self-play (Stimpson *et al.*, 2001)

The generalised prisoner's dilemma (GPD)

The generalised prisoner's dilemma (GPD)

- Generalised payoff matrix

	C	D
C	σ, σ	$0, 1$
D	$1, 0$	δ, δ

Reward payoff: σ Sucker payoff: 0
Temptation payoff: 1 Punishment payoff: δ

The generalised prisoner's dilemma (GPD)

- Generalised payoff matrix

	C	D
C	σ, σ	$0, 1$
D	$1, 0$	δ, δ

Reward payoff:	σ	Sucker payoff:	0
Temptation payoff:	1	Punishment payoff:	δ

Constraints: $0 < \delta < \sigma < 1$ and $1/2 < \sigma$.

The generalised prisoner's dilemma (GPD)

- Generalised payoff matrix

	C	D
C	σ, σ	$0, 1$
D	$1, 0$	δ, δ

Reward payoff:	σ	Sucker payoff:	0
Temptation payoff:	1	Punishment payoff:	δ

Constraints: $0 < \delta < \sigma < 1$ and $1/2 < \sigma$. (Why?)

The generalised prisoner's dilemma (GPD)

- Generalised payoff matrix

	C	D
C	σ, σ	$0, 1$
D	$1, 0$	δ, δ

Reward payoff: σ Sucker payoff: 0
Temptation payoff: 1 Punishment payoff: δ

Constraints: $0 < \delta < \sigma < 1$ and $1/2 < \sigma$. (Why?)

- Use Karandikar *et al.*'s algorithm.

The generalised prisoner's dilemma (GPD)

- Generalised payoff matrix

	C	D
C	σ, σ	$0, 1$
D	$1, 0$	δ, δ

Reward payoff:	σ	Sucker payoff:	0
Temptation payoff:	1	Punishment payoff:	δ

Constraints: $0 < \delta < \sigma < 1$ and $1/2 < \sigma$. (Why?)

- Use Karandikar *et al.*'s algorithm.
  - States for satisficing play:

The generalised prisoner's dilemma (GPD)

- Generalised payoff matrix

	C	D
C	σ, σ	$0, 1$
D	$1, 0$	δ, δ

Reward payoff:	σ	Sucker payoff:	0
Temptation payoff:	1	Punishment payoff:	δ

Constraints: $0 < \delta < \sigma < 1$ and $1/2 < \sigma$. (Why?)

- Use Karandikar *et al.*'s algorithm.
  - States for satisficing play:
    - (A_t, α_t) for the row player.

The generalised prisoner's dilemma (GPD)

- Generalised payoff matrix

	C	D
C	σ, σ	$0, 1$
D	$1, 0$	δ, δ

Reward payoff:	σ	Sucker payoff:	0
Temptation payoff:	1	Punishment payoff:	δ

Constraints: $0 < \delta < \sigma < 1$ and $1/2 < \sigma$. (Why?)

- Use Karandikar *et al.*'s algorithm.
  - States for satisficing play:
    - (A_t, α_t) for the row player.
    - (B_t, β_t) for the column player.

The generalised prisoner's dilemma (GPD)

- Generalised payoff matrix

	C	D
C	σ, σ	$0, 1$
D	$1, 0$	δ, δ

Reward payoff: σ Sucker payoff: 0
Temptation payoff: 1 Punishment payoff: δ

Constraints: $0 < \delta < \sigma < 1$ and $1/2 < \sigma$. (Why?)

- Use Karandikar *et al.*'s algorithm.
  - States for satisficing play:
    - (A_t, α_t) for the row player.
    - (B_t, β_t) for the column player.
  - The initial states are denoted by (A_0, α_0) and (B_0, β_0), respectively.

Self-play: possible dynamics

Self-play: possible dynamics

1. **Stability**.

Self-play: possible dynamics

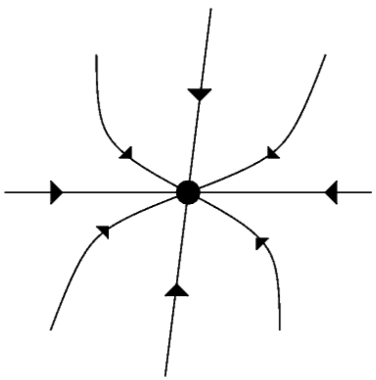
1. **Stability**. Convergence to a fixed action profile.

Self-play: possible dynamics

1. **Stability**. Convergence to a fixed action profile. This happens if and only if

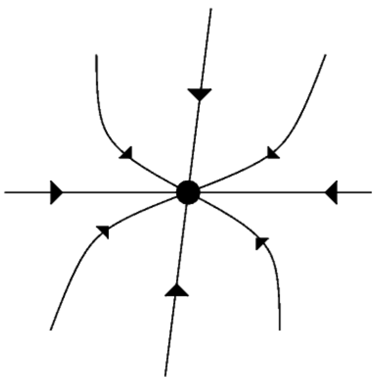
$$\alpha_t^A \leq \pi_t^A \quad \text{and} \quad \alpha_t^B \leq \pi_t^B.$$

for all $t \geq T$, for some $T \geq 0$.



Self-play: possible dynamics

1. **Stability**. Convergence to a fixed action profile. This happens if and only if



$$\alpha_t^A \leq \pi_t^A \quad \text{and} \quad \alpha_t^B \leq \pi_t^B.$$

for all $t \geq T$, for some $T \geq 0$.

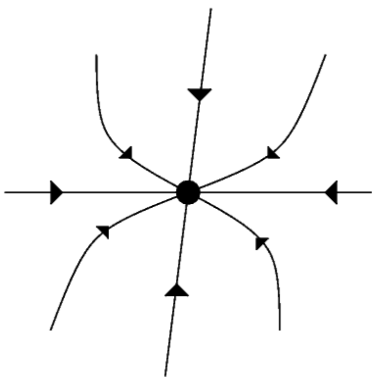
2. **Periodicity**.

Self-play: possible dynamics

1. **Stability**. Convergence to a fixed action profile. This happens if and only if

$$\alpha_t^A \ \leq \ \pi_t^A \quad \text{and} \quad \alpha_t^B \ \leq \ \pi_t^B.$$

for all $t \geq T$, for some $T \geq 0$.



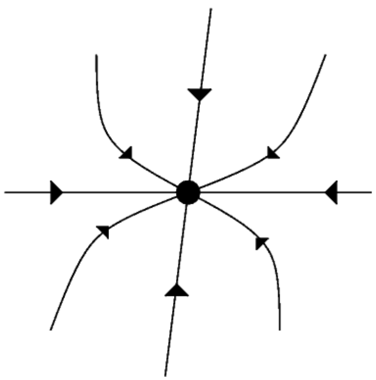
2. **Periodicity**. Convergence to a cycle of action profiles

Self-play: possible dynamics

1. **Stability**. Convergence to a fixed action profile. This happens if and only if

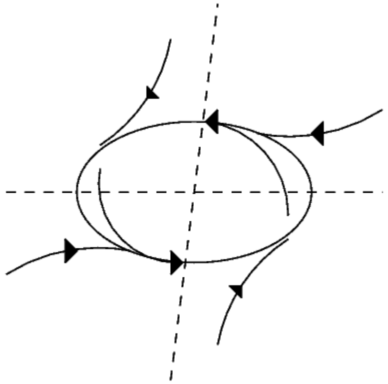
$$\alpha_t^A \leq \pi_t^A \quad \text{and} \quad \alpha_t^B \leq \pi_t^B.$$

for all $t \geq T$, for some $T \geq 0$.



2. **Periodicity**. Convergence to a cycle of action profiles, e.g.

$$(D, D),\ (D, C),\ (C, D),\ (D, D),\ (D, C),\ (C, D),\ \ldots$$

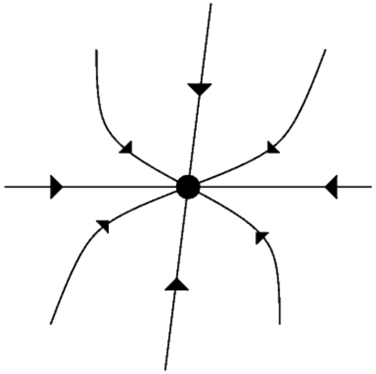


Self-play: possible dynamics

1. **Stability**. Convergence to a fixed action profile. This happens if and only if

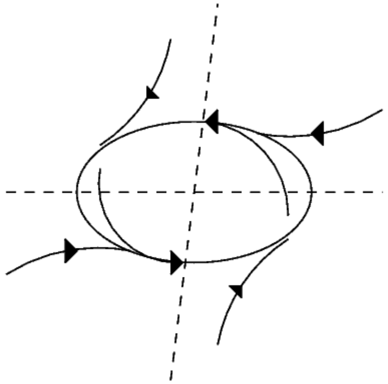
$$\alpha_t^A \leq \pi_t^A \quad \text{and} \quad \alpha_t^B \leq \pi_t^B.$$

for all $t \geq T$, for some $T \geq 0$.



2. **Periodicity**. Convergence to a cycle of action profiles, e.g.

$$(D, D),\ (D, C),\ (C, D),\ (D, D),\ (D, C),\ (C, D),\ \ldots$$



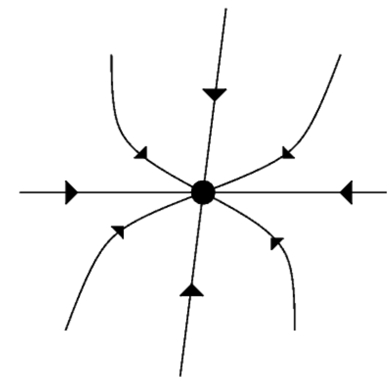
3. **Chaos**.

Self-play: possible dynamics

1. **Stability**. Convergence to a fixed action profile. This happens if and only if

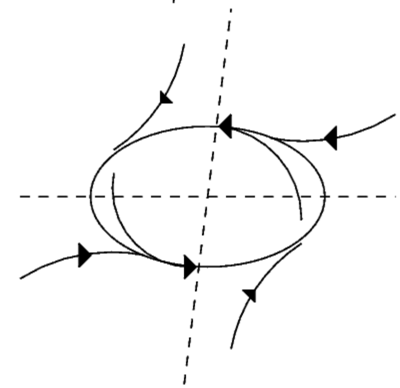
$$\alpha_t^A \leq \pi_t^A \quad \text{and} \quad \alpha_t^B \leq \pi_t^B.$$

for all $t \geq T$, for some $T \geq 0$.

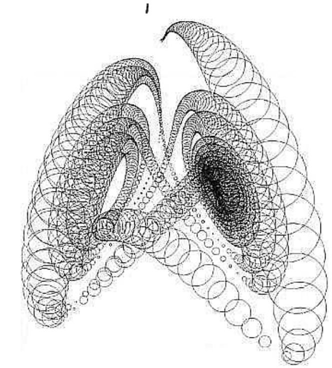


2. **Periodicity**. Convergence to a cycle of action profiles, e.g.

$$(D,D),\ (D,C),\ (C,D),\ (D,D),\ (D,C),\ (C,D),\ \ldots$$



3. **Chaos**. Deterministic but non-periodic behaviour.



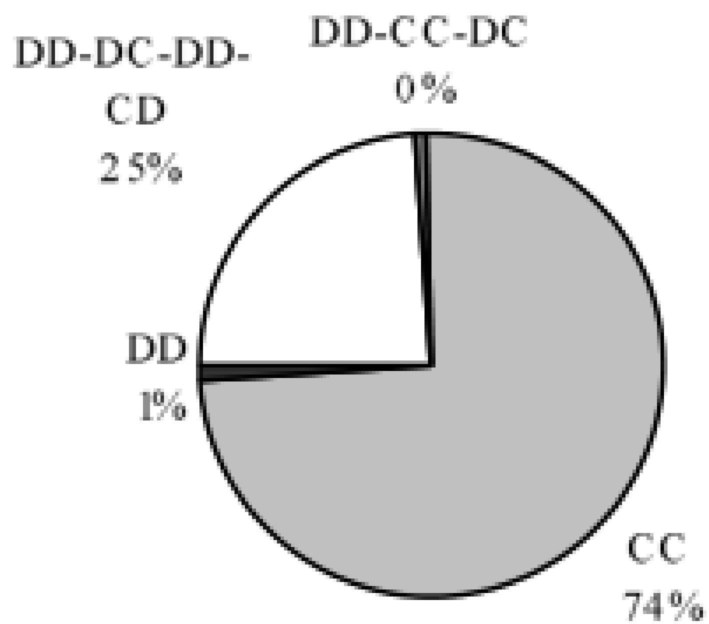
Experiments throughout the parameter space

Parameter space

	Symbol	Min	Max
Reward payoff	σ	0.51	1.0
Punishment payoff	δ	0.1	σ
Initial aspirations	α_0, β_0	0.5	2.0
Initial actions	A_0, B_0	50% C, 50% D	
Persistence rate	λ	0.1	0.9

Table 1: Distribution of parameters for simulations.

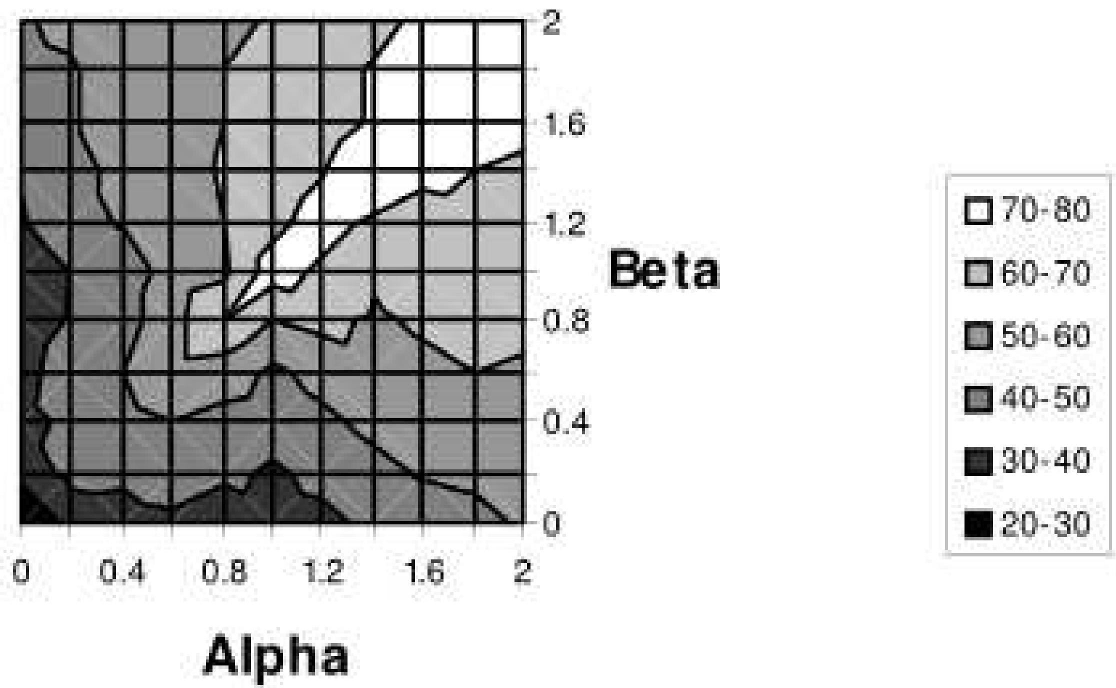
Frequencies of each of the possible outcomes



Frequencies of each of the possible outcomes from 5,000 trials. Parameters were randomly selected as described in Table 1.

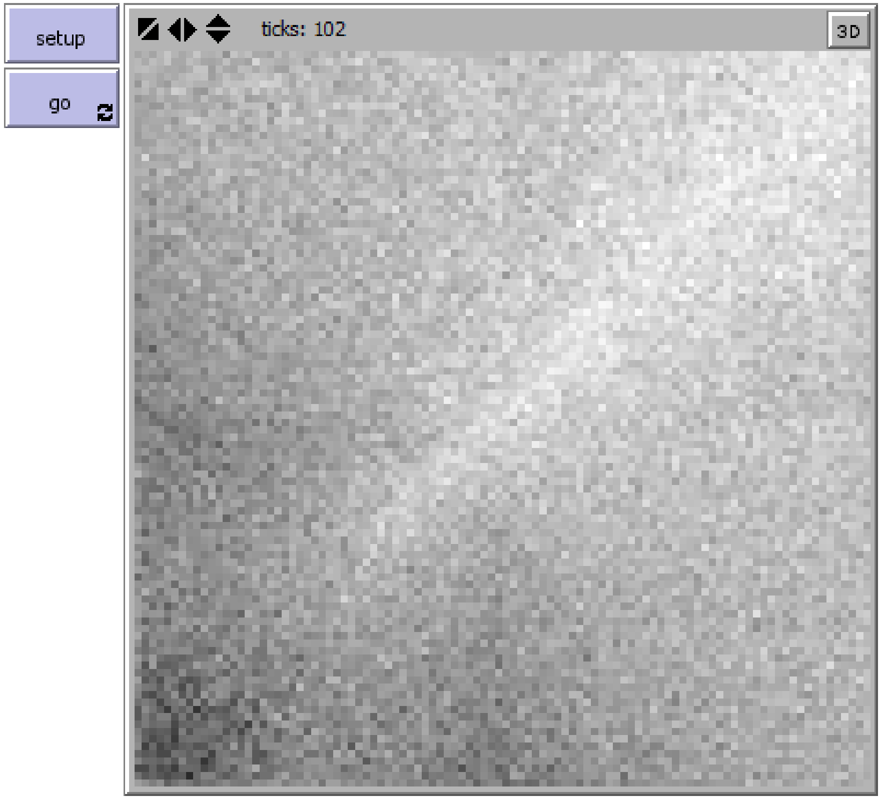
(From: "Satisficing and Learning Cooperation in the Prisoner's Dilemma", Stimpson *et al.*, 2001.)

Mutual cooperation as a result of initial aspirations



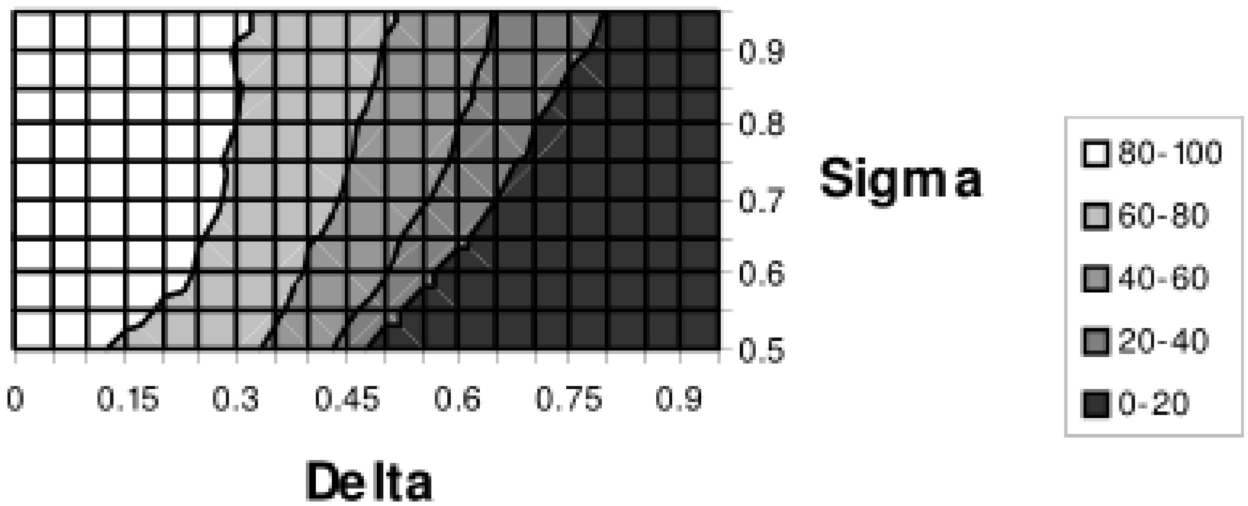
A contour plot of the percentage of trials out of 1,000 that converged to mutual cooperation as a function of initial aspirations. Light colors indicate that in most of the trials with the given initial aspirations, the agents learned to cooperate. Parameters other than α_0 and β_0 were randomly selected from Table 1. (From: Stimpson *et al.*, 2001.)

Same experiment with Netlogo



A Netlogo plot of the percentage of trials out of 100 that converged to mutual cooperation as a function of initial aspirations. Light colors indicate that in most of the trials the agents learned to cooperate. Parameters other than α_0 and β_0 were randomly selected from Table 1.

Mutual cooperation as a result of reward and punishmen



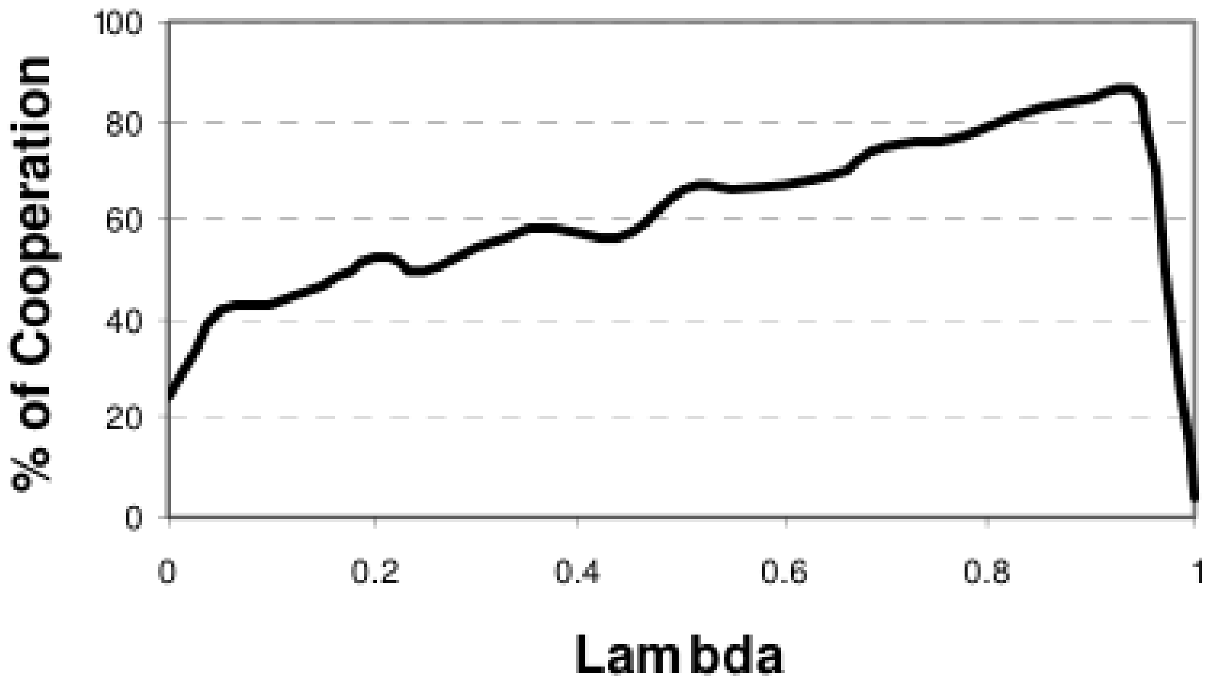
A contour plot of the percentage of trials out of 1,000 that converged to mutual cooperation as a function of each (δ, σ) pair. Light colors indicate that most of the trials converged to mutual cooperation. Parameters other than δ and σ were randomly selected from Table 1. (From: Stimpson *et al.*, 2001.)

Effects of the initial actions

Initial actions	Cooperation
Random	73.7%
CC	81.6%
DD	81.6%
CD or DC	66.7%

Table 2: Percentage of cooperation out of 1,000 trials as a function of initial actions. Parameters other than A_0 and B_0 were randomly selected from Table 1. (From: "Satisficing and Learning Cooperation . . . ", Stimpson *et al.*, 2001.)

Effect of the persistence rate

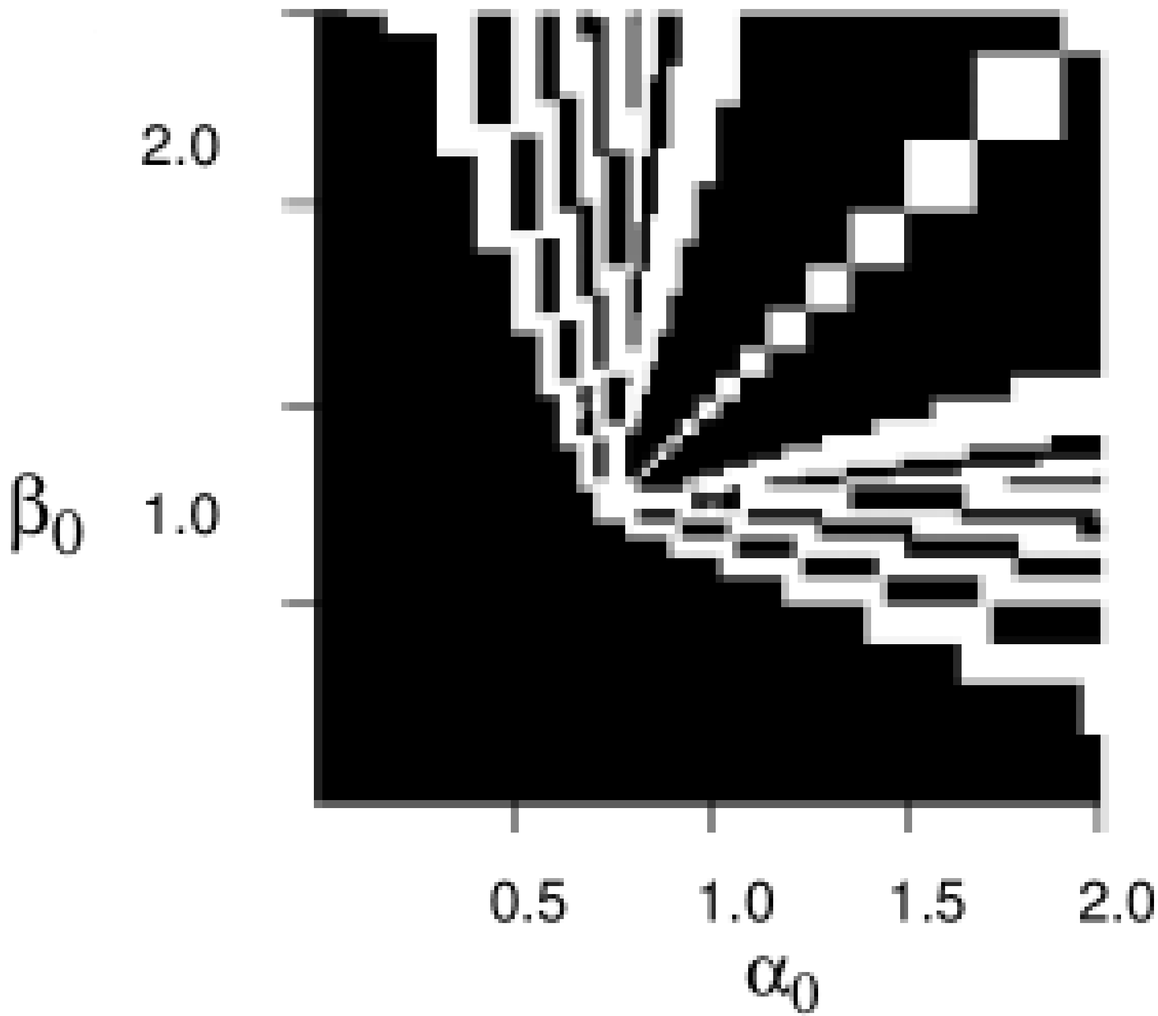


Percentage of trials out of 1,000 that converged to mutual cooperation as a function of the persistence rate, λ. Parameters other than λ were selected randomly as described in Table 1.

(From: "Satisficing and Learning Cooperation …", Stimpson *et al.*, 2001.)

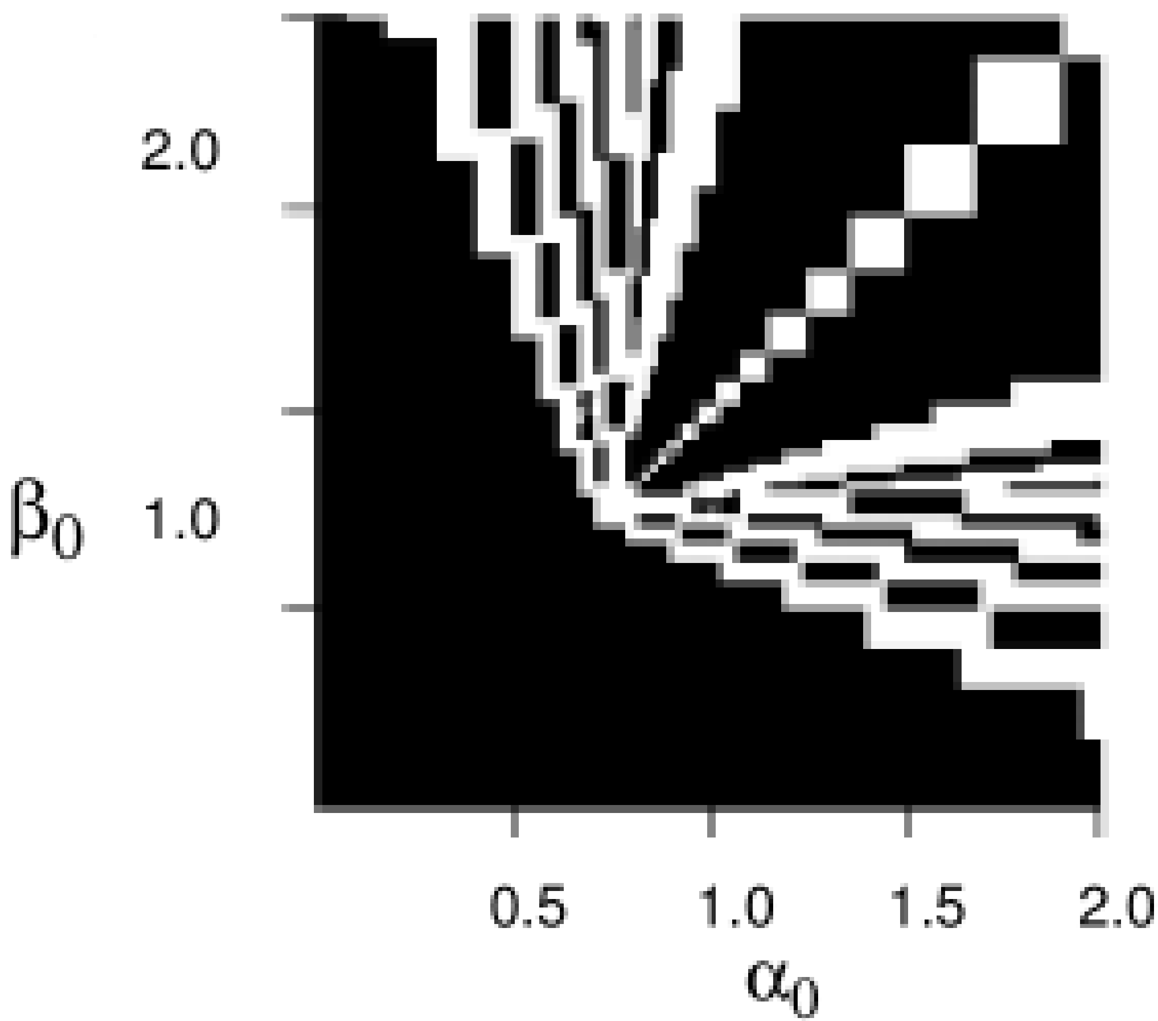
Experiments with specific parameters

Final outcome as a result of initial aspirations



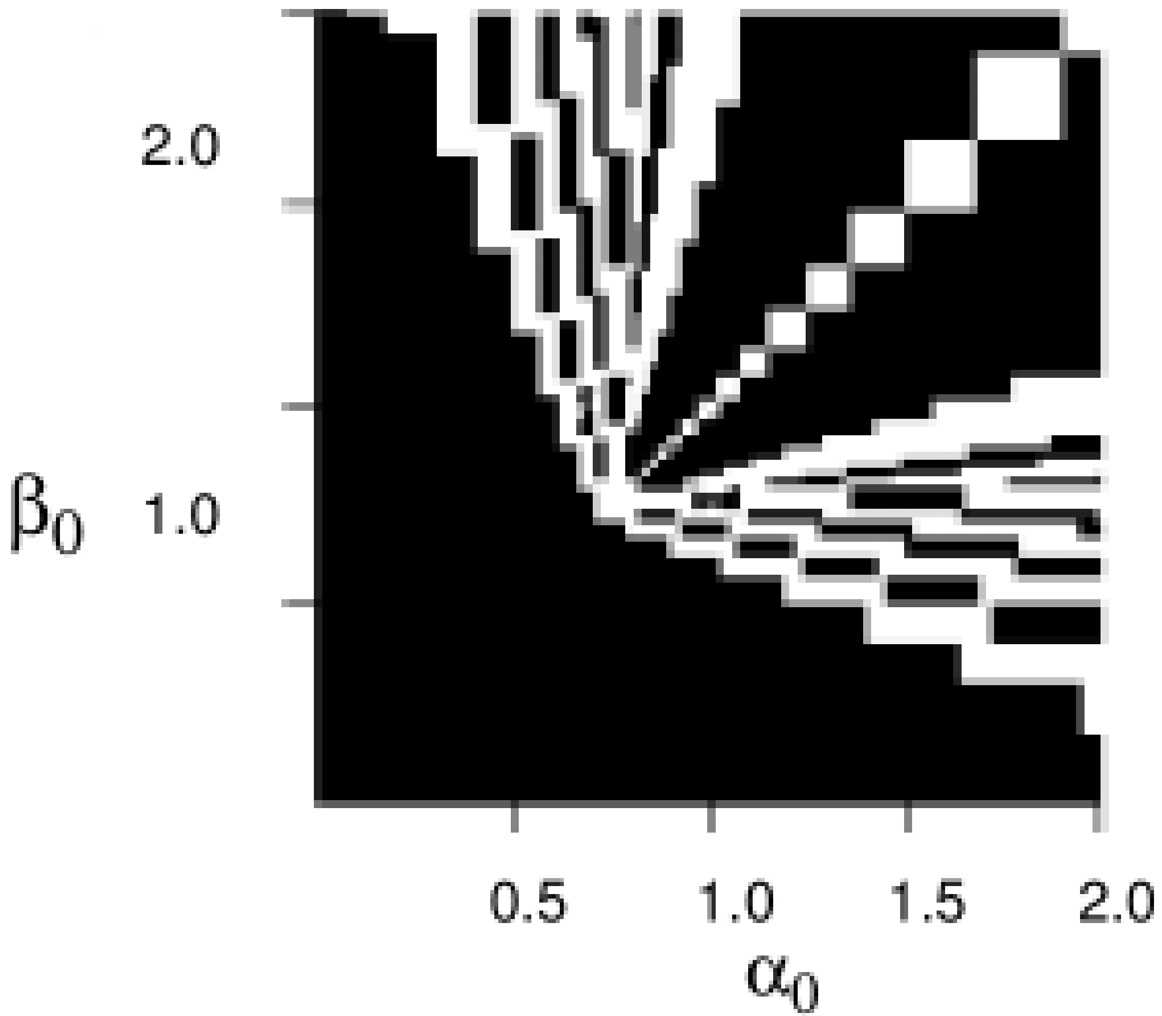
Final outcome as a result of initial aspirations

- Initial aspiration of player A on x-axis; Initial aspiration of player B on y-axis.



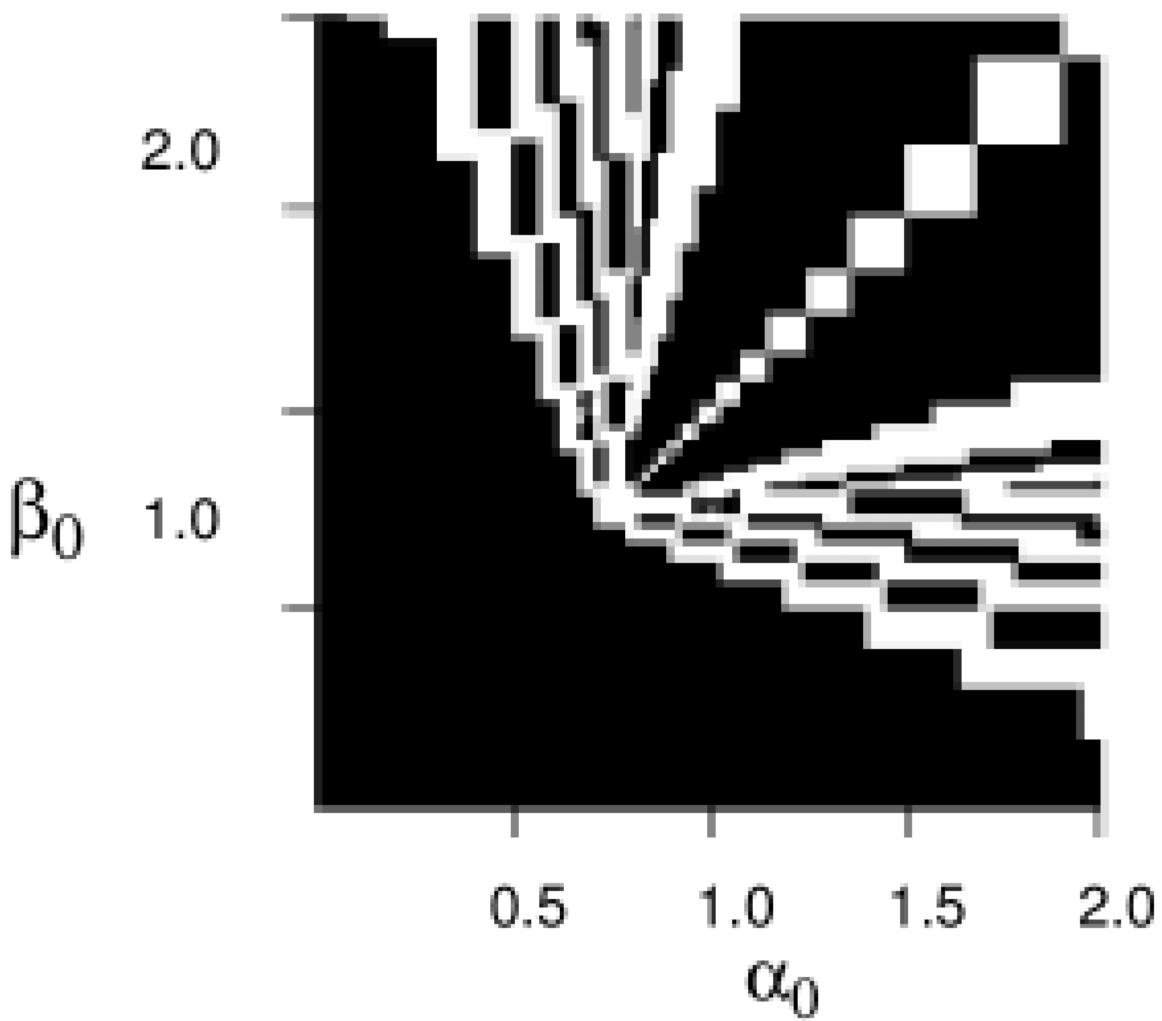
Final outcome as a result of initial aspirations

- Initial aspiration of player A on x-axis; Initial aspiration of player B on y-axis.
- White: convergence to (C, C); black: convergence to (D, D); grey: periodic or chaotic behaviour.



Final outcome as a result of initial aspirations

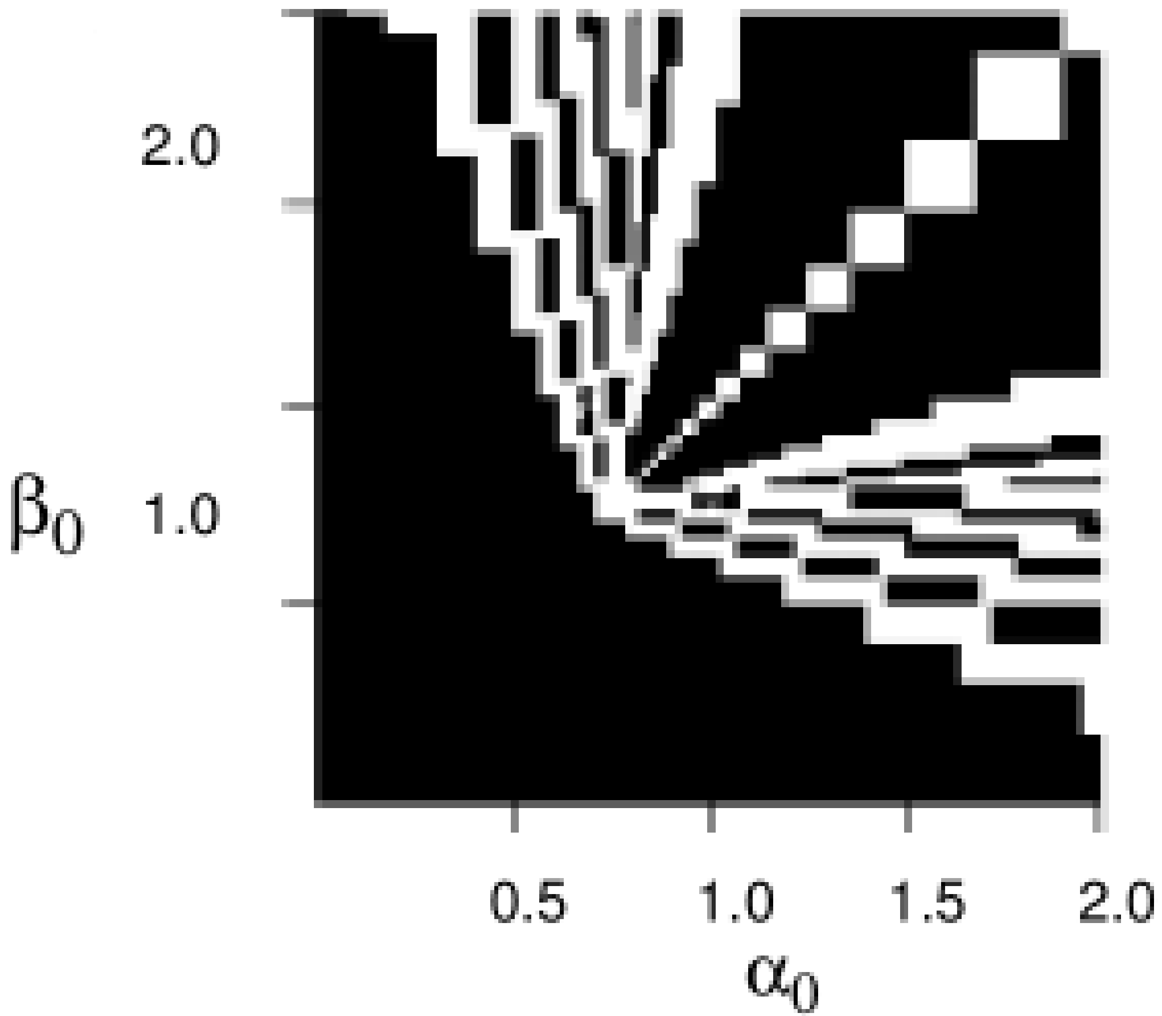
- Initial aspiration of player A on x-axis; Initial aspiration of player B on y-axis.
- White: convergence to (C, C); black: convergence to (D, D); grey: periodic or chaotic behaviour.
- $(A_0, B_0) = (D, D)$, $\sigma = 0.8$, $\delta = 0.7$, $\lambda = 0.9$.



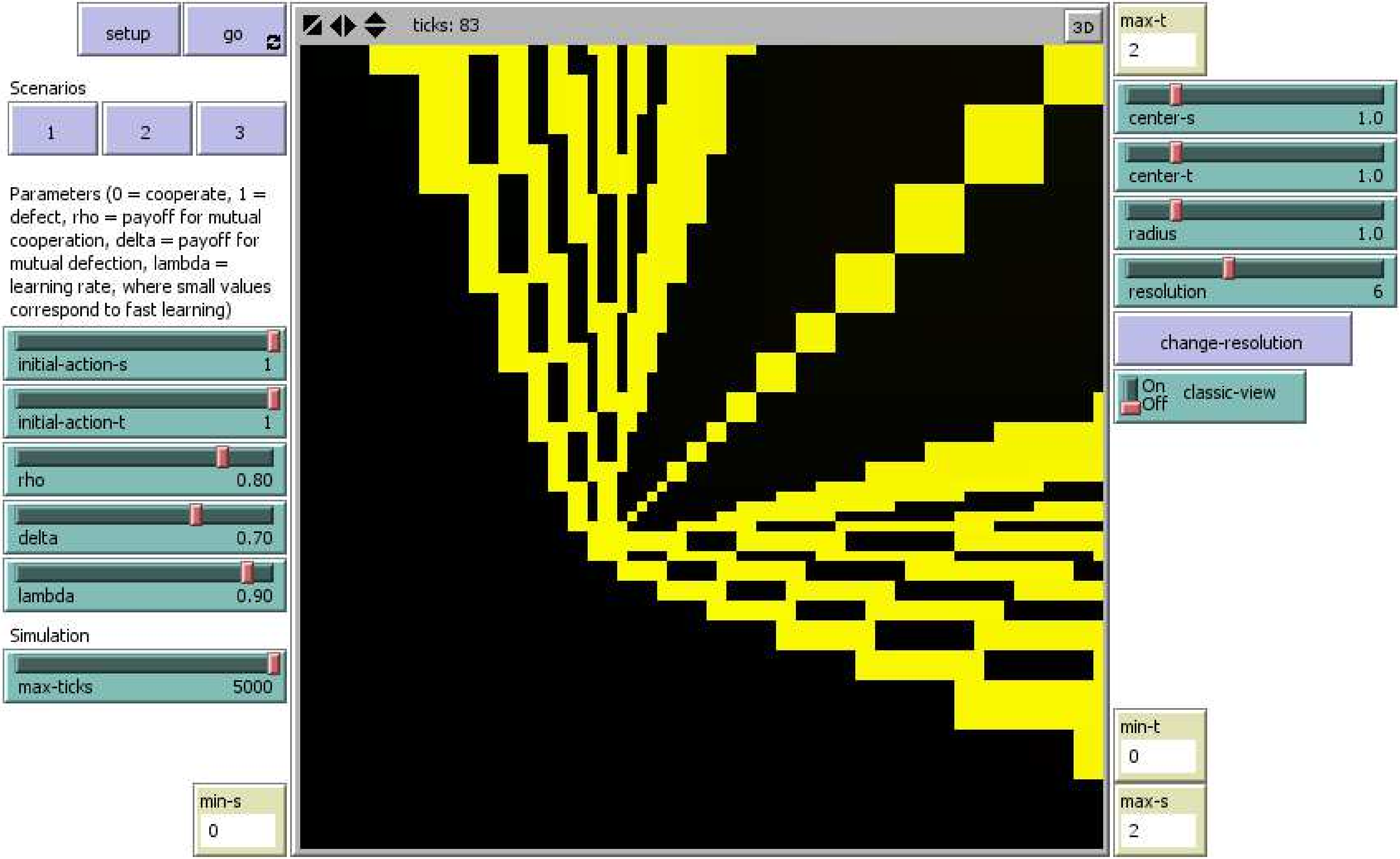
Final outcome as a result of initial aspirations

- Initial aspiration of player A on x-axis; Initial aspiration of player B on y-axis.
- White: convergence to (C, C); black: convergence to (D, D); grey: periodic or chaotic behaviour.
- $(A_0, B_0) = (D, D)$, $\sigma = 0.8$, $\delta = 0.7$, $\lambda = 0.9$.

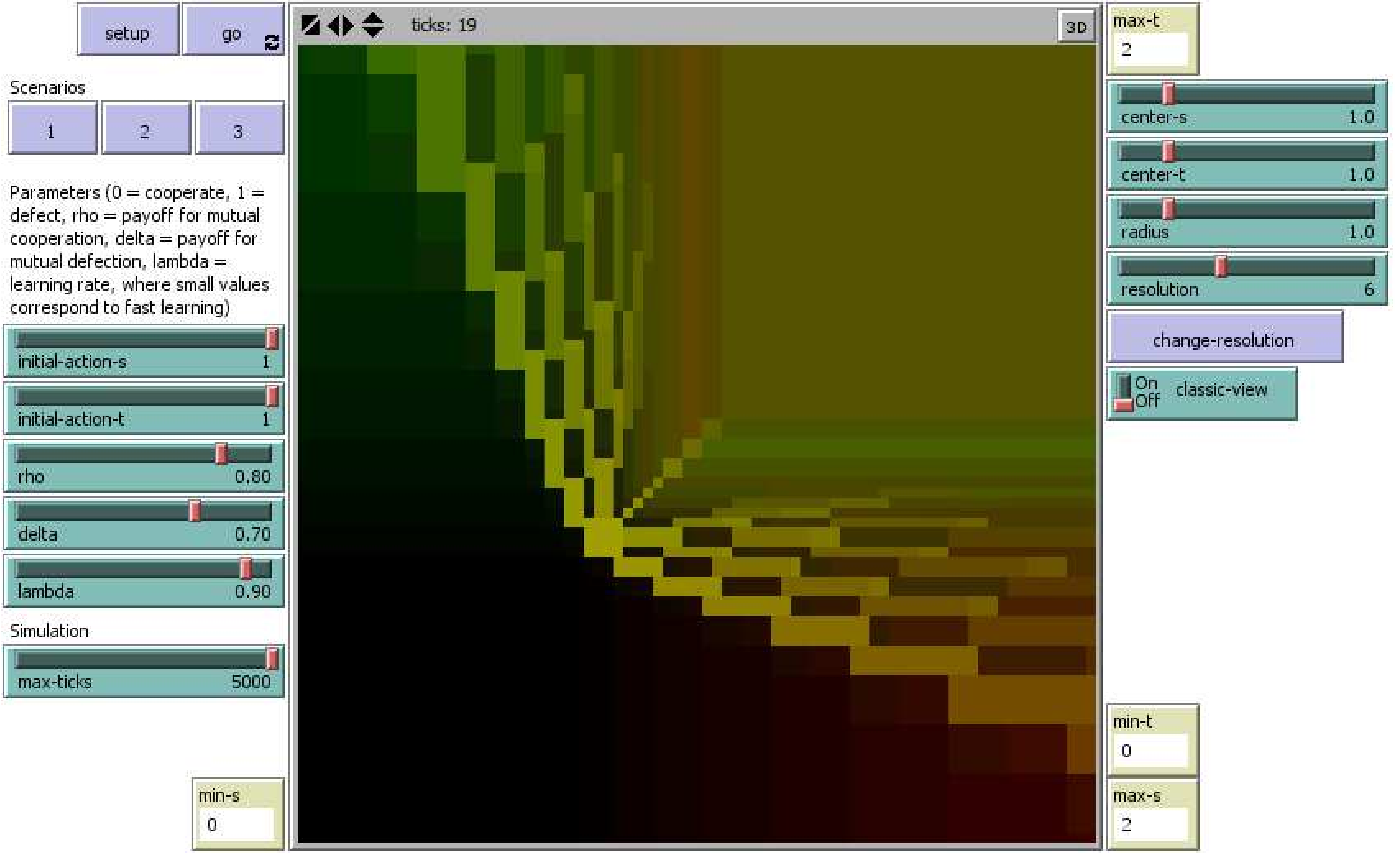
(From: "Satisficing and Learning Cooperation in the Prisoner's Dilemma", Stimpson *et al.*, 2001.)



Final outcome as a result of initial aspirations (demo)



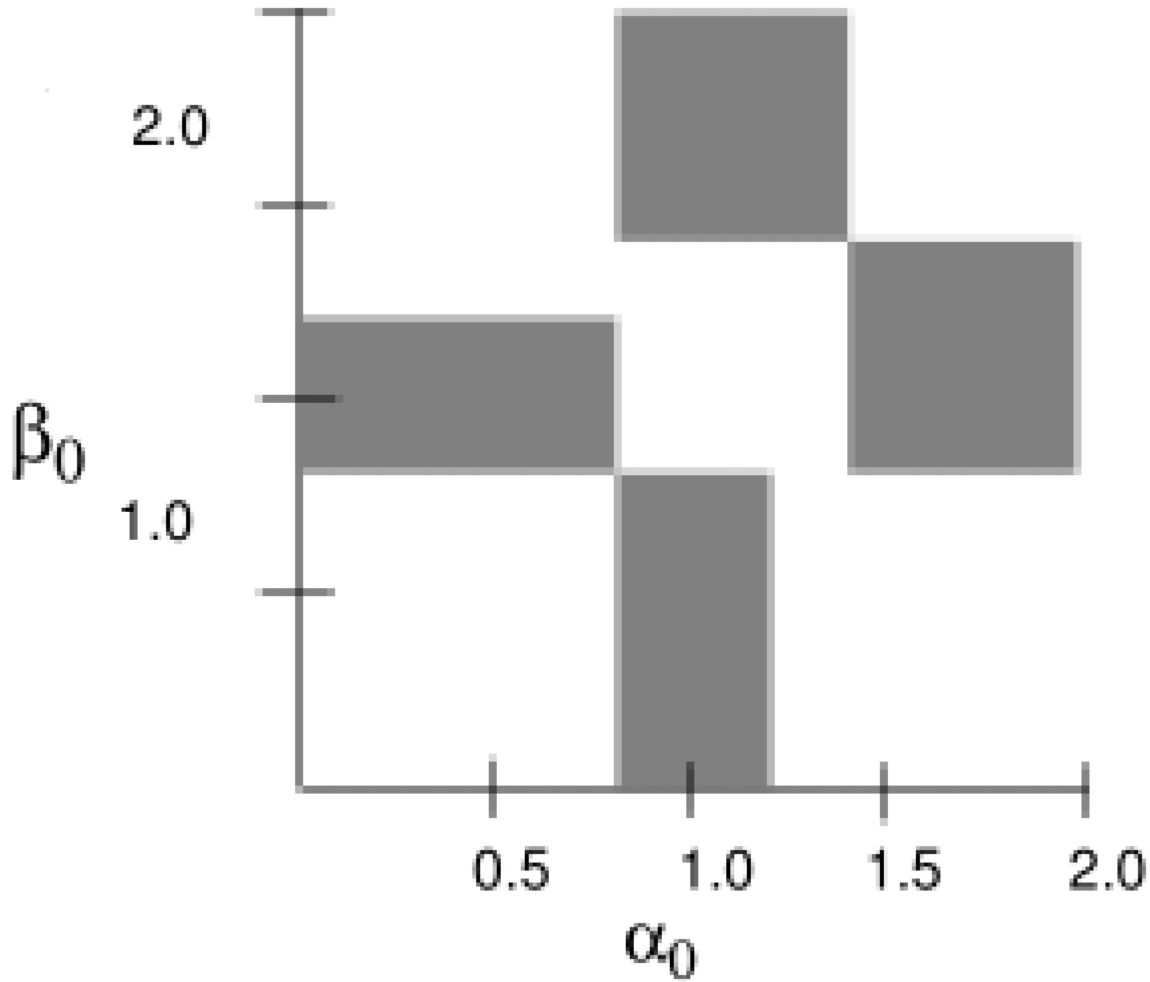
Final outcome as a result of initial aspirations (buildup)



Final outcome as a result of initial aspirations

- Initial aspiration of player A on x-axis; Initial aspiration of player B on y-axis.
- White: convergence to (C, C); black: convergence to (D, D); grey: periodic or chaotic behaviour.
- $(A_0, B_0) = (C, C)$, $\sigma = 0.8$, $\delta = 0.5$, $\lambda = 0.5$.

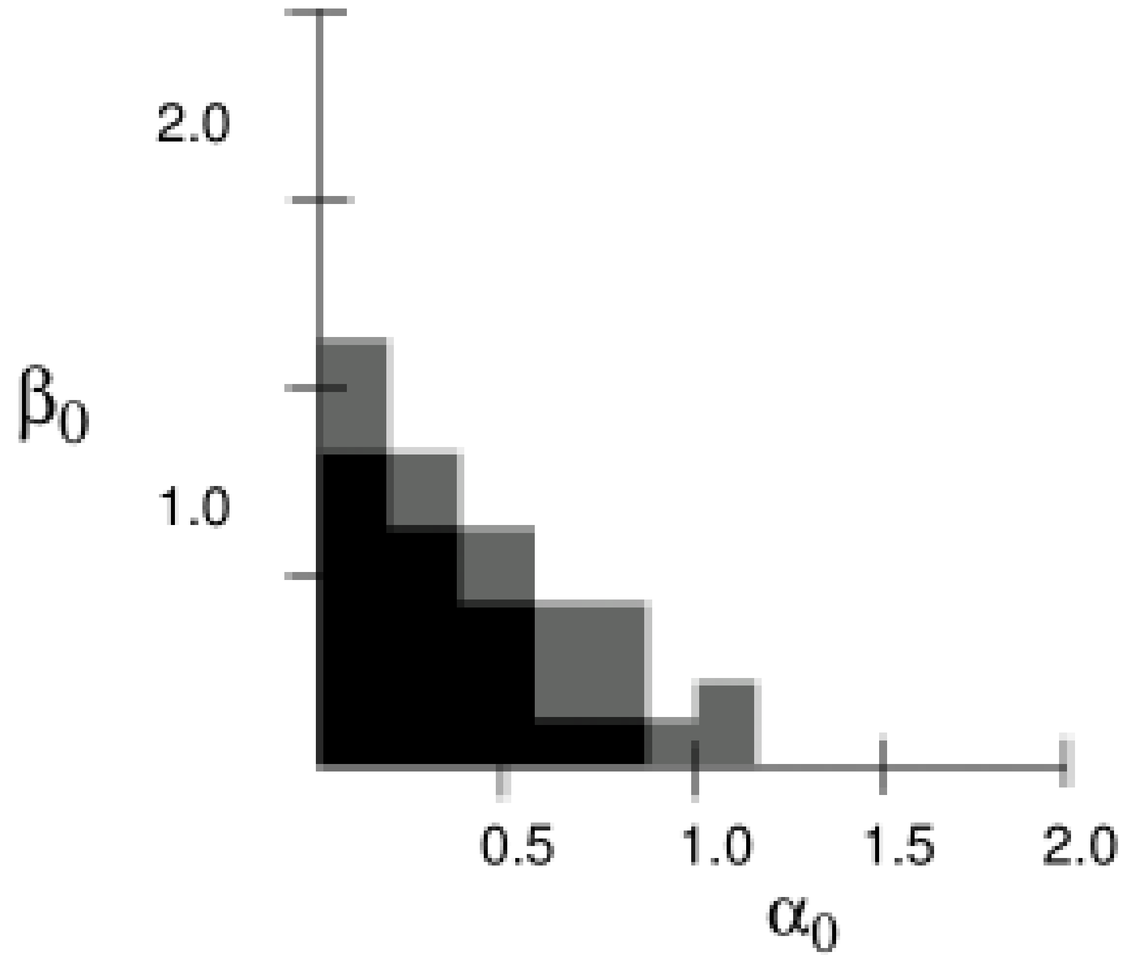
(From: "Satisficing and Learning Cooperation in the Prisoner's Dilemma", Stimpson *et al.*, 2001.)



Final outcome as a result of initial aspirations

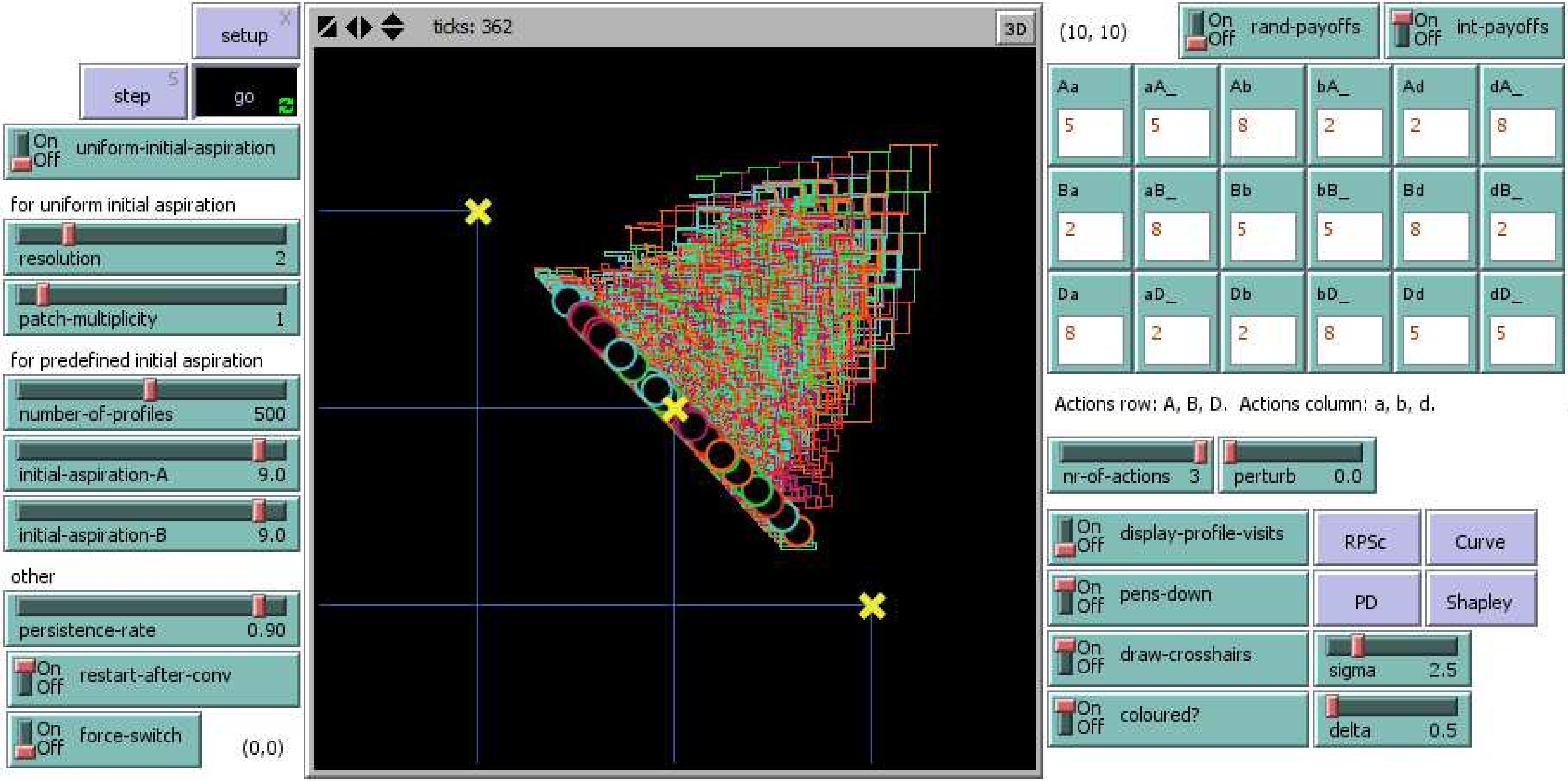
- Initial aspiration of player A on x-axis; Initial aspiration of player B on y-axis.
- White: convergence to (C, C); black: convergence to (D, D); grey: periodic or chaotic behaviour.
- $(A_0, B_0) = (D, C)$, $\sigma = 0.6$, $\delta = 0.5$, $\lambda = 0.8$.

(From: "Satisficing and Learning Cooperation in the Prisoner's Dilemma", Stimpson *et al.*, 2001.)

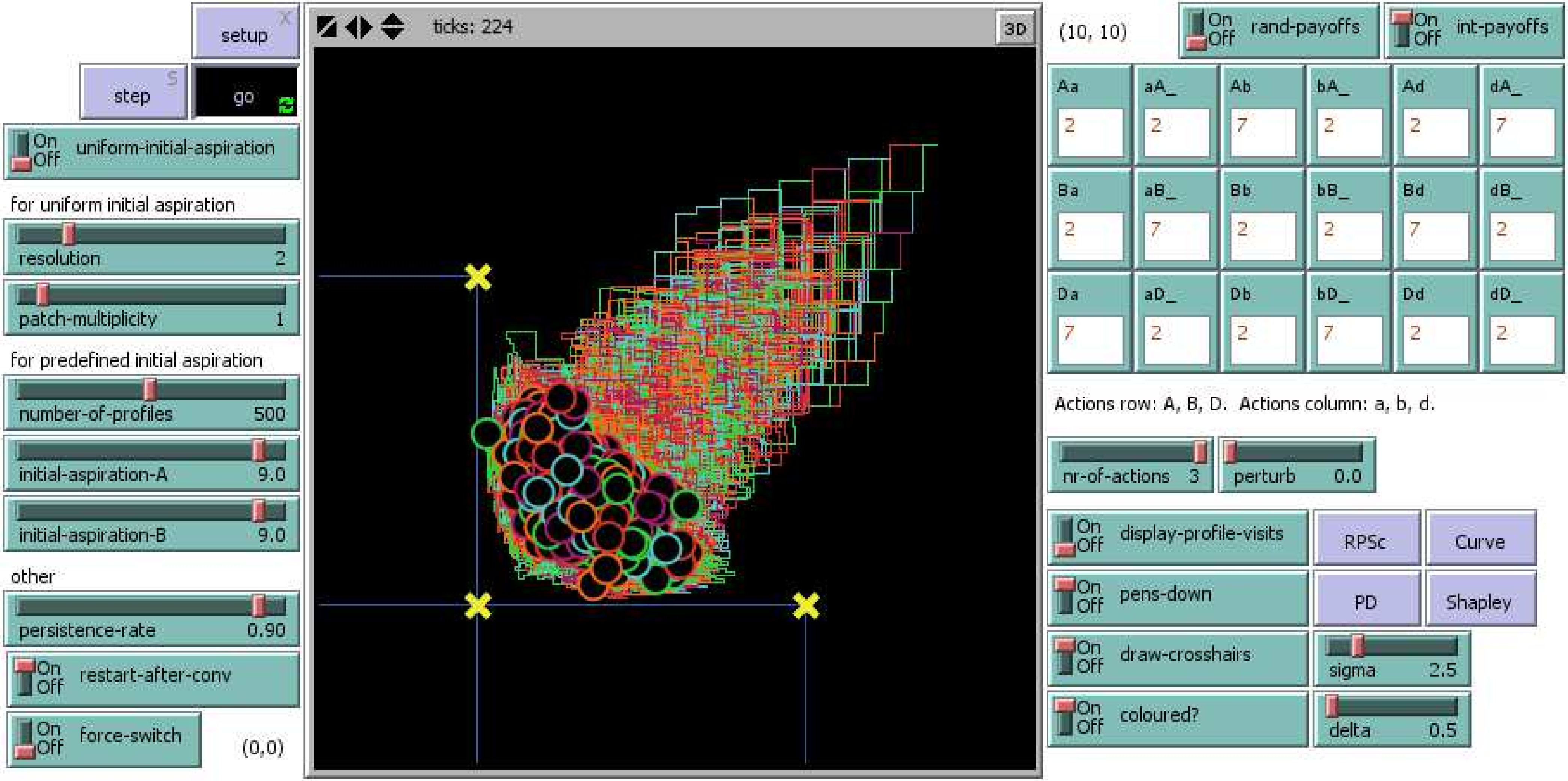


Difficult games for satisficing play

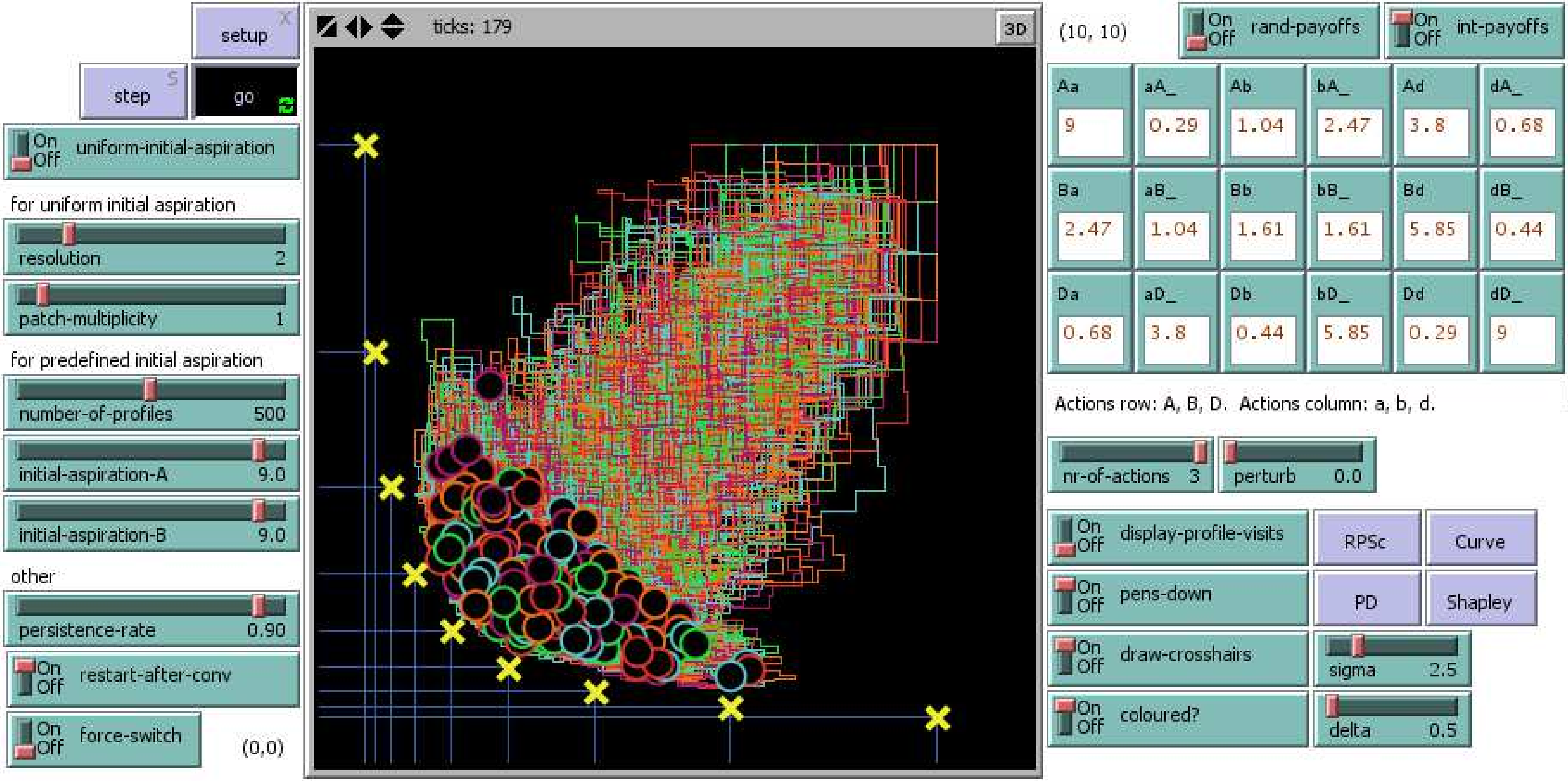
Difficult games for satisficing play (RPSc)



Difficult games for satisficing play (Shapley)



Difficult games for satisficing play (Curve)



Regret matching as a form of satisficing play

Regret matching as a form of satisficing play

Regret matching as a form of satisficing play

- Regret matching can be cast in a reinforcement rule with an aspiration level $\bar{u}^t$ (cf. Strategic Learning, H. Peyton Young, Ch. 2, p. 22).

Regret matching as a form of satisficing play

- Regret matching can be cast in a reinforcement rule with an aspiration level $\bar{u}^t$ (cf. Strategic Learning, H. Peyton Young, Ch. 2, p. 22).
- Define the reinforcement increment for every action x in round t as

$$\Delta r_x^t =_{Def} u(x, y^t) - \bar{u}^t.$$

Regret matching as a form of satisficing play

- Regret matching can be cast in a reinforcement rule with an aspiration level $\bar{u}^t$ (cf. Strategic Learning, H. Peyton Young, Ch. 2, p. 22).

- Define the reinforcement increment for every action x in round t as

$$\Delta r_x^t =_{Def} u(x, y^t) - \bar{u}^t.$$

- Define the propensities in round $t+1$ as

$$\theta_x^{t+1} =_{Def} \left[\sum_{s=1}^{t} \Delta r_x^t \right]_+$$

Regret matching as a form of satisficing play

- Regret matching can be cast in a reinforcement rule with an aspiration level $\bar{u}^t$ (cf. Strategic Learning, H. Peyton Young, Ch. 2, p. 22).
- Define the reinforcement increment for every action x in round t as

$$\Delta r_x^t =_{Def} u(x, y^t) - \bar{u}^t.$$

- Define the propensities in round $t + 1$ as

$$\theta_x^{t+1} =_{Def} \left[\sum_{s=1}^{t} \Delta r_x^t \right]_+$$

- This is like standard reinforcement, but now **all** actions in a given period are reinforced, whether or not they are actually played.

Regret matching as a form of satisficing play

- Regret matching can be cast in a reinforcement rule with an aspiration level $\bar{u}^t$ (cf. Strategic Learning, H. Peyton Young, Ch. 2, p. 22).
- Define the reinforcement increment for every action x in round t as

$$\Delta r_x^t =_{Def} u(x, y^t) - \bar{u}^t.$$

- Define the propensities in round $t+1$ as

$$\theta_x^{t+1} =_{Def} \left[\sum_{s=1}^{t} \Delta r_x^t \right]_+$$

- This is like standard reinforcement, but now **all** actions in a given period are reinforced, whether or not they are actually played.
- Hypothetical reinforcement takes into account virtual payoffs. (Payoffs that never materialised.)

Regret matching as a form of satisficing play

- Regret matching can be cast in a reinforcement rule with an aspiration level $\bar{u}^t$ (cf. Strategic Learning, H. Peyton Young, Ch. 2, p. 22).
- Define the reinforcement increment for every action x in round t as

$$\Delta r_x^t =_{Def} u(x, y^t) - \bar{u}^t.$$

- Define the propensities in round $t+1$ as

$$\theta_x^{t+1} =_{Def} \left[\sum_{s=1}^{t} \Delta r_x^t \right]_+$$

- This is like standard reinforcement, but now **all** actions in a given period are reinforced, whether or not they are actually played.
- Hypothetical reinforcement takes into account virtual payoffs. (Payoffs that never materialised.)

  The vector Δr^t is a vector of virtual reinforcements—gains or losses relative to the current average that that *would have* materialised if a given action x had been played at time t.

Conclusions

Conclusions

- Agents should have high enough initial aspirations.

Conclusions

- Agents should have high enough initial aspirations.
- Agents should learn, but slowly.

Conclusions

- Agents should have high enough initial aspirations.
- Agents should learn, but slowly.
- The difference between payoffs for mutual defection and mutual cooperation should be maximized.

Conclusions

- Agents should have high enough initial aspirations.
- Agents should learn, but slowly.
- The difference between payoffs for mutual defection and mutual cooperation should be maximized.
- Agents should start out with similar behavior.

Conclusions

- Agents should have high enough initial aspirations.
- Agents should learn, but slowly.
- The difference between payoffs for mutual defection and mutual cooperation should be maximized.
- Agents should start out with similar behavior.

As a test, a final set of 5,000 simulations was run within the following confined parameter space:

Parameter	**Min**	**Max**

Conclusions

- Agents should have high enough initial aspirations.
- Agents should learn, but slowly.
- The difference between payoffs for mutual defection and mutual cooperation should be maximized.
- Agents should start out with similar behavior.

As a test, a final set of 5,000 simulations was run within the following confined parameter space:

Parameter	**Min**	**Max**
α_0, β_0	σ	2.0

Conclusions

- Agents should have high enough initial aspirations.
- Agents should learn, but slowly.
- The difference between payoffs for mutual defection and mutual cooperation should be maximized.
- Agents should start out with similar behavior.

As a test, a final set of 5,000 simulations was run within the following confined parameter space:

Parameter	**Min**	**Max**
α_0, β_0	σ	2.0
λ	0.8	1.0

Conclusions

- Agents should have high enough initial aspirations.
- Agents should learn, but slowly.
- The difference between payoffs for mutual defection and mutual cooperation should be maximized.
- Agents should start out with similar behavior.

As a test, a final set of 5,000 simulations was run within the following confined parameter space:

Parameter	**Min**	**Max**
α_0, β_0	σ	2.0
λ	0.8	1.0
σ	0.51	1.0

Conclusions

- Agents should have high enough initial aspirations.
- Agents should learn, but slowly.
- The difference between payoffs for mutual defection and mutual cooperation should be maximized.
- Agents should start out with similar behavior.

As a test, a final set of 5,000 simulations was run within the following confined parameter space:

Parameter	**Min**	**Max**
α_0, β_0	σ	2.0
λ	0.8	1.0
σ	0.51	1.0
δ	0.1	$\sigma - 0.4$

Conclusions

- Agents should have high enough initial aspirations.
- Agents should learn, but slowly.
- The difference between payoffs for mutual defection and mutual cooperation should be maximized.
- Agents should start out with similar behavior.

As a test, a final set of 5,000 simulations was run within the following confined parameter space:

Parameter	**Min**	**Max**
α_0, β_0	σ	2.0
λ	0.8	1.0
σ	0.51	1.0
δ	0.1	$\sigma - 0.4$
A_0, B_0	$A_0 = B_0$	

Conclusions

- Agents should have high enough initial aspirations.
- Agents should learn, but slowly.
- The difference between payoffs for mutual defection and mutual cooperation should be maximized.
- Agents should start out with similar behavior.

As a test, a final set of 5,000 simulations was run within the following confined parameter space:

Parameter	**Min**	**Max**
α_0, β_0	σ	2.0
λ	0.8	1.0
σ	0.51	1.0
δ	0.1	$\sigma - 0.4$
A_0, B_0	$A_0 = B_0$	

Result: 100% mutual cooperation.

What next?

Gradient dynamics:

- Like fictitious play, players model (or assess) each other through mixed strategies.
- Strategies are not played, only maintained.
- Due to CKR (common knowledge of rationality, cf. Hargreaves Heap & Varoufakis, 2004), all models of mixed strategies are correct. (I.e., $q^{-i} = s^{-i}$, for all i.)
- Players gradually adapt their mixed strategies through hill-climbing in the payoff space.

What next?

Bayesian play:

Gradient dynamics:

- Like fictitious play, players model (or assess) each other through mixed strategies.
- Strategies are not played, only maintained.
- Due to CKR (common knowledge of rationality, cf. Hargreaves Heap & Varoufakis, 2004), all models of mixed strategies are correct. (I.e., $q^{-i} = s^{-i}$, for all i.)
- Players gradually adapt their mixed strategies through hill-climbing in the payoff space.

What next?

Bayesian play:

- With fictitious play, the behaviour of opponents is modelled by a single mixed strategy.

Gradient dynamics:

- Like fictitious play, players model (or assess) each other through mixed strategies.
- Strategies are not played, only maintained.
- Due to CKR (common knowledge of rationality, cf. Hargreaves Heap & Varoufakis, 2004), all models of mixed strategies are correct. (I.e., $q^{-i} = s^{-i}$, for all i.)
- Players gradually adapt their mixed strategies through hill-climbing in the payoff space.

What next?

Bayesian play:

- With fictitious play, the behaviour of opponents is modelled by a single mixed strategy.
- With Bayesian play, opponents are modelled by a probability distribution over (a possibly confined set of) mixed strategies.

What next?

Bayesian play:

- With fictitious play, the behaviour of opponents is modelled by a single mixed strategy.
- With Bayesian play, opponents are modelled by a probability distribution over (a possibly confined set of) mixed strategies.

Gradient dynamics:

What next?

Bayesian play:

- With fictitious play, the behaviour of opponents is modelled by a single mixed strategy.
- With Bayesian play, opponents are modelled by a probability distribution over (a possibly confined set of) mixed strategies.

Gradient dynamics:

- Like fictitious play, players model (or assess) each other through mixed strategies.

What next?

Bayesian play:

- With fictitious play, the behaviour of opponents is modelled by a single mixed strategy.
- With Bayesian play, opponents are modelled by a probability distribution over (a possibly confined set of) mixed strategies.

Gradient dynamics:

- Like fictitious play, players model (or assess) each other through mixed strategies.
- Strategies are not played, only maintained.

What next?

Bayesian play:

- With fictitious play, the behaviour of opponents is modelled by a single mixed strategy.
- With Bayesian play, opponents are modelled by a probability distribution over (a possibly confined set of) mixed strategies.

Gradient dynamics:

- Like fictitious play, players model (or assess) each other through mixed strategies.
- Strategies are not played, only maintained.
- Due to CKR (common knowledge of rationality, cf. Hargreaves Heap & Varoufakis, 2004), all models of mixed strategies are correct. (I.e., $q^{-i} = s^{-i}$, for all i.)

What next?

Bayesian play:

- With fictitious play, the behaviour of opponents is modelled by a single mixed strategy.
- With Bayesian play, opponents are modelled by a probability distribution over (a possibly confined set of) mixed strategies.

Gradient dynamics:

- Like fictitious play, players model (or assess) each other through mixed strategies.
- Strategies are not played, only maintained.
- Due to CKR (common knowledge of rationality, cf. Hargreaves Heap & Varoufakis, 2004), all models of mixed strategies are correct. (I.e., $q^{-i} = s^{-i}$, for all i.)
- Players gradually adapt their mixed strategies through hill-climbing in the payoff space.