Spatial Coordination Games with Agent Turnover

David Hagmann[1]* and Troy Tassier[2]

[1] Department of Social and Decision Sciences, Carnegie Mellon University, Pittsburgh, PA 15213, USA
dhagmann@andrew.cmu.edu
[2] Department of Economics, Fordham University, Bronx, NY 10458, USA
tassier@fordham.edu

Abstract. We study the effects of agent turnover and agent movement on equilibrium selection in spatial coordination games with Pareto dominant and risk dominant Nash equilibria. Our primary interest is in understanding how various dynamic processes influence equilibrium selection in games with multiple equilibria. We use agent based models and best response behaviors of agents to study our questions of interest. In general, as in Hagmann and Tassier (2012) we find that allowing agents to move increases the likelihood of attaining the Pareto dominant Nash equilibrium. The effects of agent turnover are more nuanced with the effects depending on the ability of agents to relocate on the lattice. Increasing the rate of turnover when movement is not allowed in the model increases the likelihood of attaining the Pareto dominant Nash equilibrium. Increasing the rate of turnover when movement is allowed in the model decreases the likelihood of attaining the Pareto dominant Nash equilibrium in some circumstances.

Keywords: coordination games, equilibrium selection, agent movement, agent-based modeling

1 Introduction

We study the attainment of equilibria in lattice based coordination games. As an example of a coordination game, consider the following $2x2$ normal form game:

		Player 2	
		X	Y
Player 1	X	a, a	b, c
	Y	c, b	d, d

Throughout the paper we assume $a > c$, $d > b$ such that there exist two pure strategy Nash Equilibria, X, X and Y, Y. Thus agents attempt to coordinate

* Corresponding Author

with their play partners on one of the two Nash equilibria. Thus our game of interest is a standard $2x2$ coordination game. Further, we assume that $a > d$ such that X, X is the Pareto dominant Nash equilibrium. Harsanyi and Selton [1] define equilibrium Y, Y to be a risk dominant Nash equilibrium if $(a-c)(a-c) < (d-b)(d-b)$ which is equivalent to $a + b < c + d$. Our primary interest in this paper will be with payoffs assigned such that Y, Y qualifies as risk dominant. We study equilibrium selection in these games using an agent-based model.

There exists a large literature on the long run selection of equilibria in these coordination games. As examples, Ellison [2]; Kandori, Mailath, and Rob [3]; and Young [4], study equilibrium selection in an evolutionary framework where agents are randomly matched with game partners. They find that the risk dominant Nash equilibrium is the unique stochastically stable equilibrium when agents have a small probability of making mistakes in strategy selection. Morris [5] studies the spread of a Pareto dominant Nash equilibrium where agents play a spatial coordination game on various topologies. He finds that a Pareto dominant equilibrium may be favored in some network based coordination games if the number of neighbors in the network expands at an intermediate rate (quickly, but not too quickly). In recent research, Hagmann and Tassier [6] add the ability of agents to move in these coordination games. They find that allowing agents to move greatly increases the likelihood of attaining the Pareto dominant Nash equilibrium.

In this paper, we introduce agent turnover in two ways. In the first, we allow agent to be *born* onto the lattice that we study. We begin with one agent and introduce new agents at various rates. We do this while also controlling the agents' ability to move and investigate how the interaction of new agent arrival rates and the ability to move affects equilibrium selection. In the second set of experiments, once new agents are introduced onto the lattice, we begin allowing some agents to "die" and new agents are born to replace them. We then study how this turnover process influences equilibrium selection results.

2 The Model

Overview

In our model, agents populate a 12x12 lattice. The number of agents in the model is less than the number of locations on the lattice so that there are some vacant locations. These agents are defined by two variables: their location on the grid and their strategy in the cooperation game described in the previous section with the following specific payoffs:

		Player 2	
		X	Y
Player 1	X	$2, 2$	$-2, 0$
	Y	$0, -2$	$1, 1$

An agent's initial location on the lattice is determined randomly. Only one agent can populate a lattice cell at any given time, and each agent can have up to eight neighbors (other agents located in an adjacent cell, including the diagonals). Agents located at the edge of the lattice have fewer neighbors (i.e. the world is not a torus).[3] Agents are introduced into the model at various rates as described below.

An agent plays one strategy against all her neighbors and calculates the average payoff. If there are no neighbors, the payoff is zero. After an agent's first round of play, she will, in some scenarios, adapt her strategy to be a best-response to the game-play of the previous round.

We study a scenario in which agents, once created, exist until the end of the simulation, and a second scenario in which turnover occurs. In this scenario, a fixed number of randomly selected agents are terminated and replaced by newly created successors. We also vary the ability of agents to move in each of these scenarios.

Processes and scheduling

The initialization procedure creates one agent at a random location and assigns strategy Y to her (the strategy that corresponds to the risk dominant Nash equilibrium.) In the first round, $k \in [1, 50]$ agents are created (spawned) at random locations, and assigned strategy X with probability $p \in [0, 1]$. Both k and p are deterministic and vary across treatments. All agents then play the coordination game against all their neighbors, using only one strategy for all matches, and calculate their average payoff. Subsequently, they test the other strategy (e.g. an agent that played X will also calculate her average payoff for Y) and compare the average payoff from playing against all neighbors. If the payoff of the alternate strategy is higher, the agent will change her strategy for the next round.

If movement is enabled (a condition set for all agents), then each agent will choose an open spot on the grid at random. At that location, she will calculate her potential payoff for both strategies. She will then compare the payoff from the best performing strategy at the new random location against the payoff from the best performing strategy at the current location. If the former is greater, she will move to the new location and adapt that strategy; otherwise, she will stay at the current location.

At that point, the current round ends and the process is repeated: new agents are created, stochastically assigned an initial strategy, and agents again play against their neighbors and, if movement is enabled, choose whether to relocate to a new spot. If agent replacement is disabled, then this process is repeated until the end of the simulation (when k additional agents do not fit on the lattice.)

With agent replacement, however, the process contains an additional component. Once the number of agents exceeds 100, a round begins by first eliminating k agents, chosen at random. Every agent has the same chance of being eliminated. Once k agents are removed, the process continues as without replacement.

[3] In our previous work with models of this type we have not found edge effects to be significant.

Termination conditions vary based on whether or not the model allows for replacement. In treatments where replacement does not take place, the simulation ends once there are no longer enough open spots on the lattice to create k new agents. With replacement, the simulation ends after a total of 300 rounds.[4]

3 Results

We present the results of four sets of simulations. First, we model an information cascade where an agent initially determines a best-response strategy and plays that strategy throughout the simulation. We explore what happens with and without movement. Next, we allow agents to change strategies and compare directly the rate of cooperation with and without movement if only one agent is born per round. Then we show what happens to cooperation as this rate increases, both with and without movement. Finally, we terminate randomly selected agents and initialize new ones in their place. The results reported in the figures and tables below are averages based on 5,000 runs for each set of parameters and treatments.

3.1 Movement in an Information Cascade

We first model an information cascade, where agents can choose between playing strategy X, corresponding to the Pareto dominant Nash equilibrium, and strategy Y, corresponding to the Risk dominant Nash equilibrium. We consider the former to be the superior good in the classical setup. We investigate whether allowing agents to move increases coordination on X.

The setup is as follows. In every round, one new agent is created. If the agent has no neighbors at her current location, she plays strategy X with some probability $P(x)$, which we vary across experiments. If an agent has a neighbor, she plays the strategy with the highest payoff given the neighbors' strategies. Once the strategy is so determined, the agent continues to play it for the remainder of the simulation. We show the results for increases in $P(x)$ from 10% to 90% in increments of 10%.

We observe that allowing for movement significantly decreases the number of agents that play the Pareto dominant Nash Equilibrium. The difference is small for low probabilities of playing strategy X without neighbors, with 14% and 17% of agents playing the Pareto dominent Nash equilibrium with and without movement, respectively. As $P(x)$ increases, so does the difference in cooperation. If agents without neighbors play strategy X 90% of the time, then 79% of agents end up playing strategy X without movement, whereas only 59% do with movement. We simulated the experiment for spawn rates up to 10 agents per period, but did not find any differences in cooperation.

[4] We find that 300 rounds are sufficient to generate stable behavior.

Table 1. Spawn rate = 1 agent per period. Movement decreases the likelihood of attaining the Pareto dominant Nash equilibrium.

probGreen	spawnRate	coop move = false	coop move = true	similarity move = false	similarity move = true
10	1	4.41%	5.34%	95.70%	96.44%
20	1	10.17%	9.73%	91.51%	94.30%
30	1	16.84%	13.94%	87.94%	92.44%
40	1	24.25%	18.38%	85.08%	90.51%
50	1	33.51%	23.20%	83.06%	88.54%
60	1	43.79%	29.24%	82.22%	86.41%
70	1	55.05%	36.20%	82.89%	84.35%
80	1	67.08%	46.01%	85.10%	82.49%
90	1	79.18%	59.03%	88.98%	81.67%

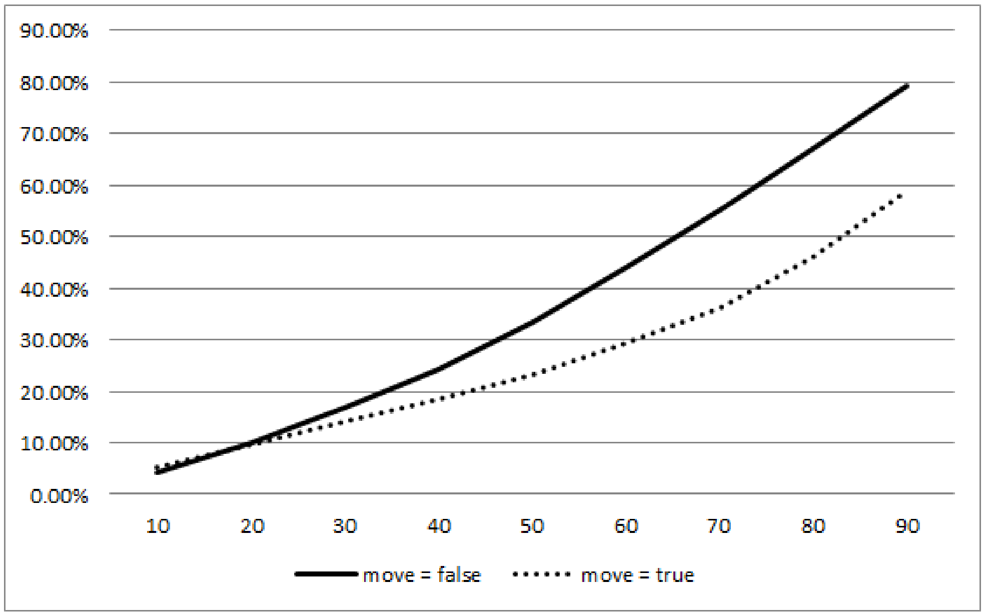

Fig. 1. Spawn rate = 1 agent per period. Movement decreases the likelihood of attaining the Pareto dominant Nash equilibrium.

3.2 Movement in the Simple Model

We begin with the simplest of our models that allow agents to change strategies. Here we ask if movement increases attainment of the Pareto dominant Nash equilibrium when the first agent picks the strategy corresponding to the risk dominant Nash equilibrium.

The setup is as follows. The first agent plays strategy Y, and in each subsequent round, one new agent is created (spawn rate = 1 agent per period). This new agent initially plays strategy X (corresponding to the Pareto dominant Nash equilibrium) with a probability $P(x)$ which we vary across experiments. The table and chart below compare the percentage of agents coordinating on the Pareto dominant Nash equilibrium for $P(x)$ in increments of 10%.

Table 2. Spawn rate = 1 agent per period. Movement greatly increases the likelihood of attaining the Pareto dominant Nash equilibrium.

P(x)	move = false	move = true
100	81.06%	99.96%
90	28.25%	99.82%
80	7.16%	99.60%
70	1.60%	98.53%
60	0.20%	96.34%
50	0.04%	92.04%
40	0.00%	84.86%
30	0.00%	71.44%
20	0.00%	54.83%
10	0.00%	28.63%

We find that movement significantly increases the likelihood of coordinating on the Pareto dominant Nash equilibrium. Without movement, the Pareto-dominant Nash equilibrium is achieved in 81% of runs if all newly created agents begin by playing strategy X. However, even a small probability of playing strategy Y greatly decreases the likelihood of attaining the Pareto dominant Nash equilibrium. If $P(x) = 0.9$ the Pareto dominant Nash equilibrium is achieved only 28% of the time. If $P(x) < 70\%$ the Pareto dominant Nash equilibrium is almost never attained without movement. The initial agent playing strategy Y has a significant and long lasting effect in this scenario.

With movement, however, the initial play by the first agent has a much smaller effect. If $P(x) \geq 0.5$ over 90% of runs result in the Pareto dominant Nash equilibrium when movement is allowed. And even when new agents play strategy X only 10% of the time, over 1/4 of runs end with the Pareto dominant Nash equilibrium. Movement has a very strong effect leading toward the Pareto dominant Nash equilibrium.

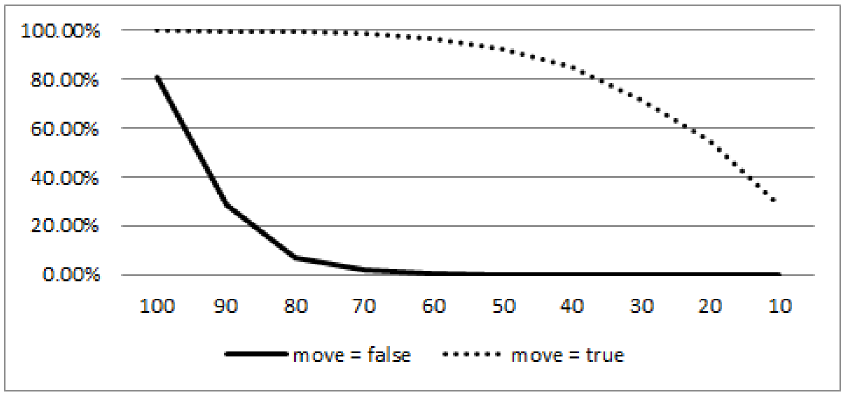

Fig. 2. Spawn rate = 1 agent per period. Movement greatly increases the likelihood of attaining the Pareto dominant Nash equilibrium.

3.3 Varying the Spawn Rate

Our second set of simulations examines the effect of rate at which new agents are introduced. We look at whether creating more agents in every round influences the share of agents playing strategy X. To do so, we spawn 1, 2, 5, and 10 agents per round. Again, we do so both without and with movement. The following charts represent averages for 5,000 iterations for each set of parameters.

Table 3. Without movement increasing the spawn rate increases the likelihood of attaining the Pareto dominant Nash equilibrium.

P(x)	spawnRate = 1	spawnRate = 2	spawnRate = 5	spawnRate = 10
100	81.06%	83.86%	90.92%	94.51%
90	28.25%	35.06%	51.83%	68.34%
80	7.16%	11.04%	23.43%	39.13%
70	1.60%	2.48%	8.50%	18.14%
60	0.20%	0.51%	2.41%	6.76%
50	0.04%	0.08%	0.74%	2.26%
40	0.00%	0.01%	0.08%	0.63%
30	0.00%	0.01%	0.01%	0.14%
20	0.00%	0.00%	0.01%	0.03%
10	0.00%	0.00%	0.00%	0.01%

We find that increasing the rate of introduction increases the likelihood of attaining the Pareto dominant Nash equilibrium when movement is not allowed. As an example, when the probability of playing X, $P(x)$, by new agents is 90% and one new agent is introduced each period, only 28% of runs results in the

Table 4. Without movement increasing the spawn rate increases the likelihood of attaining the Pareto dominant Nash equilibrium.

P(x)	spawnRate = 1	spawnRate = 2	spawnRate = 5	spawnRate = 10
100	99.96%	99.96%	100.00%	100.00%
90	99.82%	99.90%	99.79%	99.92%
80	99.60%	99.50%	98.92%	98.16%
70	98.53%	98.10%	95.33%	89.96%
60	96.34%	94.05%	85.71%	69.10%
50	92.04%	88.19%	68.42%	42.04%
40	84.86%	77.23%	45.68%	20.41%
30	71.44%	60.75%	23.79%	7.66%
20	54.83%	39.38%	10.05%	2.70%
10	28.63%	18.30%	2.69%	0.58%

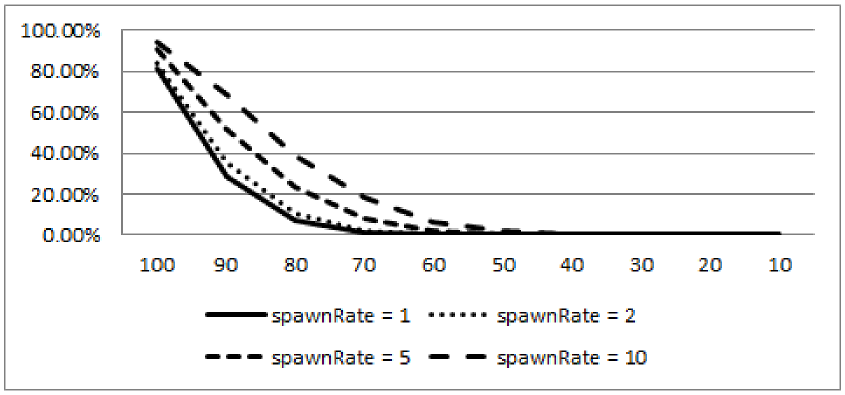

Fig. 3. Without movement increasing the spawn rate increases the likelihood of attaining the Pareto dominant Nash equilibrium.

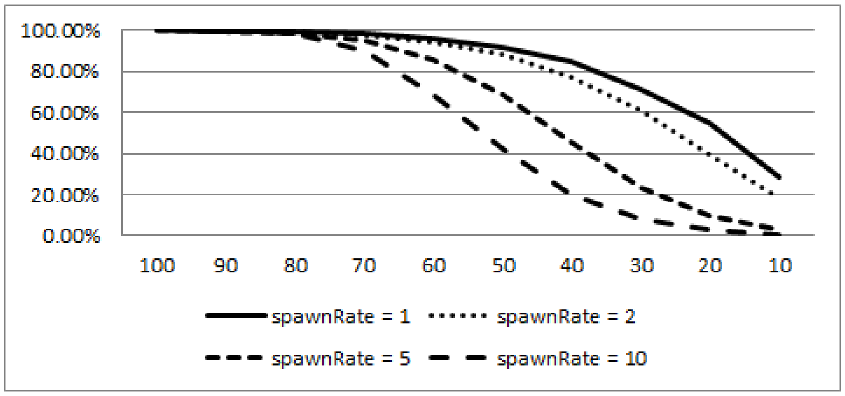

Fig. 4. When movement is allowed, increasing the spawn rate decreases the likelihood of attaining the Pareto dominant Nash equilibrium.

Pareto dominant Nash equilibrium. If the rate of introduction is increased to 10 agents per period, 60% of runs result in the Pareto dominant Nash equilibrium. However, if $P(x)$ is not sufficiently large, then increases in the rate of introduction has an insignificant effect on the likelihood of attaining the Pareto dominant Nash equilibrium. Conversely, for low values of P(x) and when agents are allowed to move, we observe a significant and non-linear decrease in the rate of cooperation with increasing spawn rates. For large values of P(x) ($P(x) > 0.8$), agents are able to coordinate on the Pareto dominant Nash equilibrium in almost all runs. But for intermediate and low levels of $P(x)$ increasing the spawn rate decreases the likelihood of attaining the Pareto dominant Nash equilibrium. As an example, at $P(x) = 0.4$ and a spawn rate of 1 agent per period, just under 85% of runs result in coordination on the Pareto dominant Nash equilibrium. But, with a spawn rate of 10 new agents per period, the likelihood of coordinating on the Pareto dominant Nash equilibrium drops to 20%. With movement and a high spawn rate, it appears that the likelihood of attaining the Pareto dominant Nash equilibrium is closely tied to the probability of new agents playing strategy X. With lower spawn rates, movement strongly favors the Pareto dominant Nash equilibrium as long as $P(x)$ is not very low.

3.4 Rates of replacement

Finally, we introduce agent replacement in our third set of simulations. The setup follows that of the previous subsection with one difference. Once the model is populated by at least 100 agents, k agents will be replaced in every round. We report averages based on 1,000 iterations for each set of parameters both with and without movement.

Similar to the previous section, if movement is not allowed, increasing the spawn rate increases the likelihood of attaining the Pareto dominant Nash equi-

Table 5. Agent replacement: Without movement increasing the spawn rate increases the likelihood of the Pareto dominant Nash equilibrium.

p(X)	k = 1	k = 5	k = 10	k = 20	k = 30	k = 50
100	69.33%	70.64%	75.55%	98.09%	99.90%	100.00%
90	13.38%	8.41%	7.18%	26.03%	64.23%	98.34%
80	1.62%	0.45%	0.33%	1.82%	1.62%	87.23%
70	0.25%	0.14%	0.29%	1.06%	0.79%	33.00%
60	0.02%	0.08%	0.18%	0.65%	0.47%	10.33%
50	0.02%	0.05%	0.12%	0.47%	0.24%	4.80%
40	0.01%	0.05%	0.09%	0.27%	0.13%	2.07%
30	0.02%	0.02%	0.06%	0.14%	0.05%	0.97%
20	0.01%	0.01%	0.03%	0.06%	0.03%	0.35%
10	0.00%	0.01%	0.01%	0.03%	0.01%	0.08%

Table 6. Agent replacement: With movement increasing the spawn rate decreases the likelihood of the Pareto dominant Nash equilibrium for low $P(x)$

p(X)	k = 1	k = 5	k = 10	k = 20	k = 30	k = 50
100	99.90%	100.00%	100.00%	100.00%	100.00%	100.00%
90	100.00%	99.98%	99.97%	99.92%	99.94%	99.75%
80	99.70%	99.76%	99.91%	99.76%	99.73%	98.75%
70	98.51%	98.46%	98.98%	99.51%	95.58%	94.62%
60	98.07%	96.14%	96.25%	93.94%	39.99%	58.52%
50	93.26%	90.02%	78.60%	44.99%	0.49%	9.49%
40	86.13%	75.77%	47.82%	1.45%	0.24%	3.06%
30	74.99%	54.35%	15.09%	0.22%	0.06%	0.93%
20	58.87%	31.27%	2.30%	0.04%	0.04%	0.26%
10	30.56%	12.52%	0.01%	0.01%	0.00%	0.05%

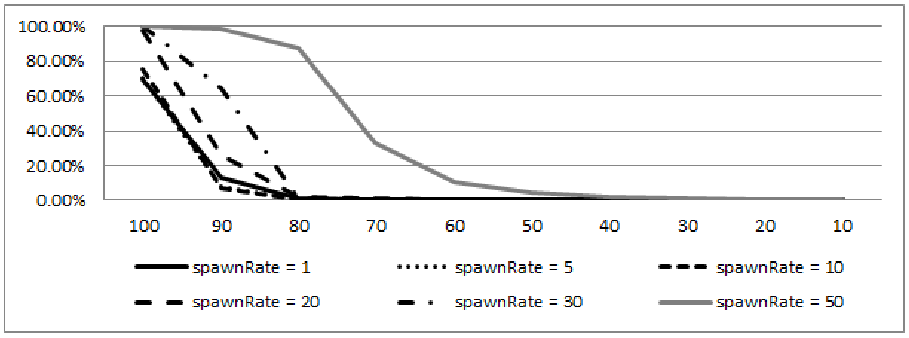

Fig. 5. Agent replacement: Without movement increasing the spawn rate increases the likelihood of the Pareto dominant Nash equilibrium.

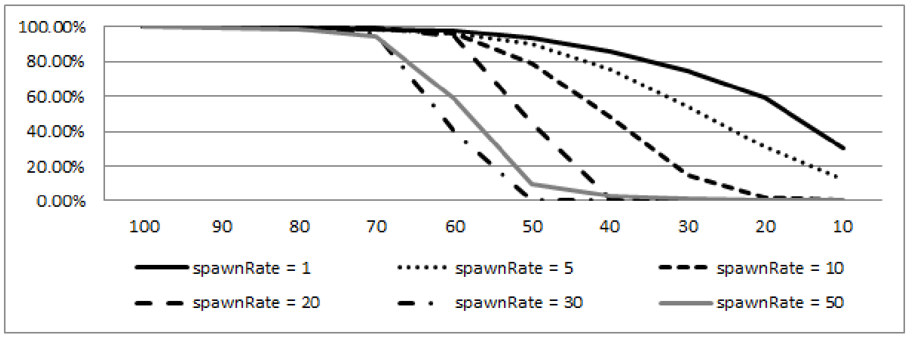

Fig. 6. Agent replacement: With movement increasing the spawn rate decreases the likelihood of the Pareto dominant Nash equilibrium for low $P(x)$.

librium, as long as $P(x)$ is large enough. For low $P(x)$, the Pareto dominant Nash equilibrium is very unlikely regardless of the spawn rate. On the other hand, with movement, increasing the spawn rate decreases the likelihood of attaining the Pareto dominant Nash equilibrium for low values of $P(x)$. As an example, for $P(x) = 0.40$ and a spawn rate of 1 ($k = 1$), 86% of runs end in the Pareto dominant Nash equilibrium. But, for a spawn rate of 50 ($k = 50$) only 3% of runs end in the Pareto dominant Nash equilibrium. For sufficiently large values of $P(x)$ with movement, almost all runs end in the Pareto dominant Nash equilibrium regardless of spawn rate.

4 Conclusion

In this paper we investigate the effect of agent turnover in a spatial coordination game. As in Hagmann and Tassier [6], we find that allowing agents to move increases the likelihood of attaining a Pareto dominant Nash equilibrium. The rate at which agents are born into the model has a complex relationship with the ability of agents to move. We find that increasing the rate of introducing new agents into the model increases the likelihood of attaining the Pareto dominant Nash equilibrium when movement is not allowed and may decrease it when movement is allowed. The complex nature of these results suggest that further investigation into agent movement and equilibrium selection is warranted.

References

1. Harsanyi, J.C. and Selten, R. (1988). A General Theory of Equilibrium Selection in Games, Cambridge, MA, MIT Press.
2. Ellison, G. (1993), Learning, Local Interaction, and Coordination. Econometrica, 61, 1047-1071.

3. Kandori, M., Mailath, G. J., and Rob, R. (1993). Learning, mutation, and long run equilibria in games. Econometrica, 61, 29-56.
4. Young, P. (1993). The Evolution of Conventions. Econometrica, 61, 57-84.
5. Morris, S. (2000). Contagion. The Review of Economics Studies, 67, 57-78.
6. Hagmann, D. and Tassier, T. (2012). Endogenous Movement and Equilibrium Selection in Spatial Coordination Games. Unpublished Manuscript.

Overview, Design Concepts, and Details

1 Purpose

The purpose of the model is to provide insights into the roles of agent movement, the rates at which agents enter the world, and agent replacement in two-by-two coordination games. The payoffs are set such that there exist both a Pareto-dominant Nash equilibrium X, X and a risk-dominant Nash equilibrium Y, Y. Agents receive a higher payoff from the Pareto-dominant Nash equilibrium, but if they play the corresponding strategy and their partner does not do so as well, they incur a loss. We consider the risk-dominant Nash equilibrium to be the inferior strategy and model how changing certain parameters influences agents' ability to coordinate on the superior equilibrium. For certain parameters of the model, we get a scenario similar to information cascades, allowing us in the future to link our findings on the role of agent movement to this rich topic.

2 Entities, state variables, and scales

The model world is a 12x12 grid. Each grid cell can only be occupied by one agent at a time. Agents represent individual actors and are all homogeneous. They are defined by two variables: their location on the lattice and the strategy they played in the last round of the coordination game (or, in the first round, the strategy they were initialized with).

3 Process overview and scheduling

Agents play a coordination game against all their neighbors. They use the same strategy for all neighbors, and calculate their average payoff from the games. An agent may have up to eight neighbors: left, right, up, down, and the diagonales. If an agent has no neighbors, her payoff is 0. After the round is completed, they calculate the payoff they had received, had they played the other strategy. If the alternate payoff is greater, they adopt that strategy for the next round; otherwise, their strategy remains unchanged. If movement is enabled, each agent (in random order) will select an open spot at random and play both strategies against all the neighbors at that new location. If the average payoff of either strategy is greater than what the agent received at the current location, she will move to the new spot and adapt the strategy with the highest average payoff. Otherwise, the agent will stay at the current location. At that point (with and without movement), new agents will be "born" according to the pre-determined spawn rate, playing strategies 1 and 2 with some given probabilities. Once these agents have spawned, agents again play their neighbors and the process repeats. If replacement is enabled, then the above process repeats until there are at least 100 agents present in the world. Once that number is reached, k agents will be terminated at random, where k is the pre-determined spawn rate. Then, k new agents are spawned using the pre-determined probabilities for strategies X and Y. The simulation ends after 300 rounds, at which point the final share of agents playing X is recorded. Without replacement, agents will continue to spawn until the grid no longer has sufficiently many open spaces to accommodate all the agents set to spawn. At that point, the simulation will terminate and record the final share of agents playing X.

4 Design concepts

4.1 Basic principles

We study the attainment of equilibria in lattice based coordination games. As an example of a coordination game, consider the following $2x2$ normal form game:

		Player 2	
		X	Y
Player 1	X	a, a	b, c
	Y	c, b	d, d

We assume $a > c$, $d > b$ such that there exist two pure strategy Nash Equilibria, X, X and Y, Y. Further, we assume that $a > d$ such that X, X is the Pareto dominant Nash equilibrium. Harsanyi and Selton [1] define equilibrium Y, Y to be a risk dominant Nash equilibrium if $(a - c)(a - c) < (d - b)(d - b)$ which is equivalent to $a + b < c + d$. Our primary interest in this paper will be with payoffs assigned such that Y, Y qualifies as risk dominant. We use the following payoff matrix in our simulations:

There exists a large literature on the long run selection of equilibria in these coordination games. As examples, Ellison [2]; Kandori, Mailath, and Rob [3]; and Young [4], study equilibrium selection in an evolutionary framework where agents are randomly matched with game partners. They find that the risk dominant Nash equilibrium is the unique stochastically stable equilibrium when agents have a small probability of making mistakes in strategy selection. Morris [5] studies the spread of a Pareto dominant Nash equilibrium where agents play a spatial coordination game on various topologies. He finds that a Pareto dominant equilibrium may be favored in some network based coordination games if the number of neighbors in the network expands at an intermediate rate (quickly, but not too quickly). In recent research, Hagmann and Tassier [6] add the ability of agents to move in these coordination games. They find that allowing agents to move greatly increases the likelihood of attaining the Pareto dominant Nash equilibrium.

In this paper, we introduce agent turnover in two ways. In the first, we allow agent to be *born* onto the lattice that we study. We begin with one agent and introduce new agents at various rates. We do this while also controlling the agents' ability to move and investigate how the interaction of new agent arrival rates and the ability to move affects equilibrium selection. In the second set of experiments, once new agents are introduced onto the lattice, we begin allowing some agents to “die” and new agents are born to replace them. We then study how this turnover process influences equilibrium selection results.

4.2 Emergence

The emergent behavior in the model is coordination on the Pareto-dominant Nash equilibrium. Intuitively, allowing agents to move to new locations could allow agents playing strategy Y to intrude in neighborhoods where agents have coordinated on X. Such an agent could then destabilize the equilibrium and lead to other agents playing Y as well. Alternatively, movement could allow agents to form pockets where sufficiently many agents play X to convert other agents. The effect of the spawn rate is also not predictable a priori. More agents decrease the length of the simulation, as the world fills up quicker, but introducing many cooperating agents at once against a single non-cooperator could quickly lead to all cooperation after the initial round. Finally, replacing agents at random could both promote or destabilize an existing equilibria.

4.3 Adaptation

Agents choose their strategy to be the best response to their neighbors’ play in the previous round. When movement is enabled, they move if they could have achieved a higher payoff at the other location using either available strategy.

4.4 Objectives

Agents seek to maximize the payoff described in the coordination game. This payoff represents some utility to them and is identical for all agents.

4.5 Prediction

When movement is allowed, agents attempt to predict their payoff at some new location. They do this by “playing” both strategies against all the neighbors of that location. If one of these strategies leads to a higher payoff than what they received at their current spot, they predict that the new location will be more rewarding and move there.

4.6 Sensing

Agents know who their neighbors are and what strategy they played last. When agents are mobile, they also know the strategies played by the neighbors at their randomly chosen new location, so that they can calculate what payoff they would have received, had they been there.

4.7 Interaction

Agents interact directly with each other when they play their strategy against their neighbors. An agent's payoff is, in turn, determined by the strategy chosen by all her neighbors. These interactions involve no communication.

4.8 Stochasticity

Agents' location is determined randomly, using a uniform distribution over all open grid cells. With the exception of the first agent, who always plays strategy Y, agents are assigned an initial strategy stochastically. The probability of being initialized with strategy X is one of the parameters varied between experiments. With mobile agents, agents sample a new location at random, with all open grid cells equally likely to be chosen. Finally, when replacement is enabled, k agents are selected at random to be removed.

4.9 Observation

At the end of the simulation, the total number of agents as well as the number of agents playing strategy X are collected. Each parameter is simulated 1,000 times (for the scenario with replacement) or 5,000 times (others). The results from all simulations are used.

5 Initialization

The model always begins with one agent who is set to play strategy Y. The agent's location is determined stochastically with a uniform distribution over all grid cells. Of interest is how this agent influences overall coordination on X, X, for example if all following agents begin by playing strategy X. The first agent selecting an inferior option is also at the heart of information cascades. We hope to expand on our findings and apply them to this field.

6 Submodels

We use the following coordination payoff matrix:

		Player 2	
		X	Y
Player 1	X	$2, 2$	$-2, 0$
	Y	$0, -2$	$1, 1$

The penalty to playing X if the other player chooses Y was set such that X, X is a Pareto dominant and Y, Y is a risk dominant Nash equilibrium. If this penalty is too great, cooperation does not occur. We set the parameter such that we can observe some cooperation over all probabilities of new agents playing strategy X. This allows us to separate the effects of increasing that probability as well as of increasing the birth rate.

When allowing agents to die, we do so once at least 100 agents have been created. We observed in our previous work (Hagmann and Tassier [6]) that the number of agents on the lattice influences the rate of cooperation with movement. When the lattice gets crowded, the effect of movement decreases as agents find it difficult to locate suitable free spots. We found a significant decrease in cooperation going from 100 agents (94.2%) to 115 agents (79.5%). Thus, we set the values such that the lattice will not be sufficiently crowded to distort our findings.

Source Code

```
turtles-own
  [ score        ;; temporary
    score-br     ;; playing best response
    score-alt-br ;; playing the alternative strategy
    score-hyp    ;; hypothetical score at a new location
    hyp-strat    ;; best-response strategy at new location
    oldxcor      ;; temporary
    oldycor      ;; temporary
  ]

to setup
   clear-all
   reset-ticks
   setup-patches
   setup-turtles
end

to go
   ;; kill and spawn turtles ;;
   if kill?
    [ if count turtles > 100 [ kill-turtles ]
      if ticks >= 300 [ stop ]
    ]
   ifelse count turtles <= (world-width * world-height -
       agents-to-spawn)
         [ spawn-turtles ]
         [ stop ]
   ;; /kill and spawn turtles ;;

   ask turtles
     [ play-br           ;; play best response to previous
         round (or initial strategy in round 1)
       play-alt-br       ;; test other strategy
       update-strategy   ;; adopt best performing strategy for
            next round
       if move?          ;; If true, look at random open patch
           , calculate payoff there, and move if it's greater
         [ move ]
     ]
  tick
end

to setup-patches
   ask patches
     [ set pcolor black ]
end
```

```
to setup-turtles
   set-default-shape turtles "person"
   ask n-of (number-of-agents) patches
     [ sprout 1 ]
   ask turtles
     [ set color red ]
   ask n-of (initial-strategy-1 / 100 * number-of-agents)
       turtles
     [ set color green ]
end

to move
   save-current-position
   find-new-spot              ;; select a random patch
   play-hyp-br                ;; play br at new location, assign
        score to score-hyp-br
   decide-move                ;; if hyp-br > br: move, otherwise
        stay
end

to play-br
   ifelse color = green       ;; green = cooperate (strategy 1),
        red = defect (strategy 2)
        [ play-strategy-1 ]
        [ play-strategy-2 ]
   set score-br score         ;; assign average payoff to score-
       br
   set score 0                ;; reset the score
end

to play-alt-br                ;; play the other strategy and
    assign payoff to score-alt-br
   ifelse color = green
        [ play-strategy-2 ]
        [ play-strategy-1 ]
   set score-alt-br score
   set score 0
end

to play-strategy-1            ;; play against all neighbors,
    average payoffs
   let n_neighbors count turtles-on neighbors
   ifelse n_neighbors != 0
        [ let neighbors-list [ self ] of turtles-on neighbors
          foreach neighbors-list
                [ let opponent ?
                  let my-color green
                  let opponent-color [ color ] of opponent
                  ifelse my-color = opponent-color
```

```
                       [ set score (score + a) ] ;; I cooperate, opponent cooperates
                       [ set score (score + b) ] ;; I cooperate, opponent defects
                ]
         set score (score / n_neighbors)         ;; take the average payoff
       ]
       [ set score 0 ]  ;; if no neighbors, the score is 0
end

to play-strategy-2
   let n_neighbors count turtles-on neighbors
   ifelse n_neighbors != 0
        [ let neighbors-list [ self ] of turtles-on neighbors
          foreach neighbors-list
                [ let opponent ?
                  let my-color red
                  let opponent-color [  color  ] of opponent
                  ifelse my-color = opponent-color
                       [ set score (score + d) ] ;; I defect, opponent defects
                       [ set score (score + c) ] ;; I defect, opponent cooperates
                ]
          set score (score / n_neighbors)
        ]
        [ set score 0 ]
end

to update-strategy
   play-alt-br
   ifelse color = green
        [ if score-alt-br > score-br [ set color red ] ]   ;; if defection is better, change to defection for next round
        [ if score-alt-br > score-br [ set color green ] ] ;; if cooperation is better, change to cooperation for next round
end

to save-current-position
   set oldxcor xcor
   set oldycor ycor
end

to find-new-spot ;; pick a random empty patch
   setxy random world-width random world-height
   move-to patch-here
   while [ any? other turtles-here ]
```

```
       [ find-new-spot ]
end

to play-hyp-br      ;; play both strategies, calculate respective score, and save strategy of better scoring response
   set hyp-strat 0 ;; reset value. If both strategies perform equally well, then this remains 0
   play-strategy-1
   let score-hyp1 score
   play-strategy-2
   let score-hyp2 score
   if score-hyp1 > score-hyp2
    [ set score-hyp score-hyp1
      set hyp-strat 1
    ]
   if score-hyp2 > score-hyp1
    [ set score-hyp score-hyp2
      set hyp-strat 2
    ]
   set score 0
end

to decide-move
   ifelse score-hyp <= score-br ;; if the new spot is not better, move back to the old spot, otherwise adopt BR for new spot
        [ setxy oldxcor oldycor
          move-to patch-here
        ]
        [ if hyp-strat = 1
           [ set color green ]
          if hyp-strat = 2
           [ set color red ]
        ]                         ;; if hyp-strat = 0, then the agent does not change color.
end

to kill-turtles
   ask n-of agents-to-spawn turtles
     [ die ]
end

to spawn-turtles
   ask n-of agents-to-spawn patches with [ not any? turtles-here ]
     [ sprout 1
            [ ifelse random 100 < probGreen
                   [ set color green ]
                   [ set color red ]
```

```
                        ]
            ]
end
```