



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Modelling and Simulating Social Systems with MATLAB

Project Report

**Influence of Success-Driven and Reputation-Based
Migration in the Evolution of Cooperation**

Byungsoo Kim & Basile Maret

Zurich
December 2015

Agreement for free-download

We hereby agree to make our source code for this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Byungsoo Kim

Basile Maret



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor .

Title of work (in block letters):

Influence of success-driven and reputation-based migration in the evolution of cooperation

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):

Kim

Maret

First name(s):

Byungsoo

Basile

With my signature I confirm that


- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work .

I am aware that the work may be screened electronically for plagiarism.

Place, date

Zürich, December 9th 2015

Signature(s)


Basile Maret

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Contents

1	Abstract	5
2	Individual contributions	5
2.1	Byungsoo Kim	5
2.2	Basile Maret	5
3	Introduction and Motivations	6
4	Description of the Model	6
4.1	Prisoner's Dilemma	6
4.2	Spatial Game with Strategies	7
5	Implementation	8
5.1	Strategy Imitation	8
5.2	Success-Driven Migration Model	9
5.3	Reputation-Based Migration Model	9
5.4	Our Model	11
5.5	Other models	11
6	Simulation Results and Discussion	12
6.1	Convergence	12
6.2	Cooperator ratios	14
6.3	Patterns	16
6.4	Discussion	18
7	Summary and Outlook	18
8	References	19

1 Abstract

Cooperation is one of the most important issue in the society in terms of the stability and productivity of the society where individuals pursue selfishness. Many studies have tried to explain the evolution of cooperation among unrelated individuals, and we can improve our understanding about it in a game-theoretical way. Most of the time, in real life, more than one factor play a role at the same time. Our model in this paper is designed to reflect that fact. It is the combination of the success-driven and reputation-based migration model to decide where to go and whether move or not *simultaneously*. As a result of comparison with existing models, our model shows that it is distinguishable from the reputation-based migration model in terms of convergence, cooperator ratios and cooperator patterns by getting the predictability of the success-driven migration model. Compared to the success-driven migration model, it gets cooperator ratio as high as the one from the success-driven migration model with a more natural way of migrating.

2 Individual contributions

2.1 Byungsoo Kim

- Implementation of the Success-Driven Migration Model
- Implementation of the Reputation-Based Migration Model
- Analysis of the Models
- Redaction of the report

2.2 Basile Maret

- Experiment of the Success-Driven Migration Model
- Experiment of the Reputation-Based Migration Model
- Analysis of the Models
- Redaction of the report

3 Introduction and Motivations

Cooperation is one of the most important issue in the society in terms of the stability and productivity of the society where individuals pursue selfishness. Many studies have tried to explain the evolution of cooperation among unrelated individuals, and we can improve our understanding about it in a game-theoretical way. Individuals are affected by interacting with neighbours so that their life is affected in either positive or negative way. Through the interaction, they notice the relative situation of their life and take an action to improve it. For instance, *imitation* of neighbour's superior strategy (be cooperator or defector) or *migration* to some other place which has a better potential benefit. There are existing models for each action.

In the *success-driven* migration model [1][2], it assumes that individuals prefer to migrate to the place where they can get the best benefit within a certain area. In an empty place, it is possible to estimate the potential benefit based on the assumption that they migrate to there and interact with their neighbours around it. Provided better potential benefit than the one of current place, individuals *immediately move* to the best candidate place.

On the other hand, individuals can maximize their future benefit by considering the reputation of their neighbours as a long-term plan, not the potential benefit of the empty sites. In the *reputation-based* migration model [3], individuals evaluate their neighbours based on their reputation. As for reputation here, a cooperator has higher reputation than a defector, and it can be accumulated over time. Contrary to the success-driven migration model, individuals *decide whether move or not* based on the neighbours' overall reputation before migrating to a random empty place.

Most of the time, in real life, more than one factor play a role at the same time. Our model in this paper is designed to reflect that fact. It is the combination of the success-driven and reputation-based migration model to decide where to go and whether move or not *simultaneously*. Lastly, we will look in detail how this model will influence the evolution of cooperation.

4 Description of the Model

4.1 Prisoner's Dilemma

We will use the well known prisoner's dilemma game in a game-theoretical way. In this game, the players can take an action among two options: either cooperate or defect. According to their behaviour, there is a payoff which motivate them to defect mutually rather than cooperate. It is represented by the payoff matrix below:

		Player 2	
		Cooperate	Defect
Player 1	Cooperate	R, R	S, T
	Defect	T, S	P, P
		In case of Prisoner's Dilemma $T > R > P > S, 2R > T$	

Figure 1: Payoff Matrix of Prisoner's Dilemma

As shown in figure 1, mutual cooperation gives reward payoff R to both players. One defection between two players gives the temptation payoff T for defector which is larger than R so that it induces the player to defect when the opponent cooperates. The opponent will get what is called the sucker's payoff S . In case of mutual defection, they will get punishment P payoff which is larger than S so that it also induces the player to defect when the opponent defect. In sum, everybody is expected to defect, called the "*tragedy of the commons*" (Hardin, 1968). We used a simplified version of the prisoner's dilemma game by setting $T = b$, $R = 1$, $P = S = 0$ [3].

4.2 Spatial Game with Strategies

We simply model social interactions on the $L \times L$ 2-dimensional spatial grid [1]. There are N individuals occupying grid sites with the ratio ρ , and they interact with m direct neighbours (von Neumann neighbourhoods). The overall payoff P of player i at iteration t is the sum of each payoffs resulting from binary interactions with all von Neumann neighbours [2], and the respective player "imitate" the strategy of best performing neighbour.

In addition, we will extend this game by including both success-driven migration and reputation-based migration. Before the imitation step, a player can move to improve their expected overall payoff to empty site within a quadratic area of $(2M + 1) \times (2M + 1)$ sites (the Moore neighbourhood of size M , e.g. the 8 neighbouring sites for $M = 1$). A player can also evaluate the surrounding environment and decide whether he leaves or not by comparing his reputation and those of his neighbours [3]. Below equations account for the reputation effect in mobility:

$$R_i(t) = \alpha R_i(t-1) + S_i(t)$$

R : Reputation

i : Individual

t : Time

α : Decaying rate (memory effect)

S : 1 if cooperate, 0 otherwise

$$P_i = \frac{R_i^\gamma}{\sum_{l \in \mathbb{N}} R_l^\gamma}$$

P : Probability of individual's leaving

γ : Mobility sensitivity

\mathbb{N} : Neighbors and itself

Figure 2: Reputation Equations

5 Implementation

The code for each model is implemented in `code/cooperation.m`, and the code for the experiment is implemented in `code/cooperation_script.m`. The initial population density ρ is 0.7, and the ration of the cooperators to the defectors is 0.5.

5.1 Strategy Imitation

Basically, all models include **Strategy Imitation** procedure as follows:

Algorithm 1 Strategy Imitation

```

1: procedure STRATEGY IMITATION( $r, c$ )
2:    $payoff_{max} \leftarrow payoff_{r,c}$ 
3:    $r_{max}, c_{max} \leftarrow r, c$ 
4:   for  $neighbour \leftarrow neighbours$  do                                 $\triangleright$  von Neumann Neighbour  $m$ 
5:     if  $payoff_{neighbour} > payoff_{max}$  then                             $\triangleright$  Find the Best Neighbour
6:        $payoff_{max} \leftarrow payoff_{neighbour}$ 
7:        $r_{max}, c_{max} \leftarrow r_{neighbour}, c_{neighbour}$ 
8:     end if
9:   end for
10:   $strategy_{r,c} \leftarrow strategy_{r_{max}, c_{max}}$                      $\triangleright$  Imitate Strategy (Cooperator or Defector)
11: end procedure

```

5.2 Success-Driven Migration Model

In success-driven migration model, each individual update its payoff and compute the fictitious payoff of empty neighbour sites, then migrate to there as follows:

Algorithm 2 Success-Driven Migration Model

```

1: procedure SUCCESS-DRIVEN MIGRATION MODEL
2:   for  $iter \leftarrow 1$  to  $iter_{max}$  do
3:     for  $r, c \leftarrow grid$  do ▷ In Random Sequential Order
4:       Update Payoff( $r, c$ )
5:        $r_{new}, c_{new} \leftarrow \text{Success-Driven Migration}(r, c)$ 
6:       Strategy Imitation( $r_{new}, c_{new}$ )
7:     end for
8:   end for
9: end procedure

1: procedure UPDATE PAYOFF( $r, c$ )
2:   for  $neighbour \leftarrow neighbours$  do ▷ von Neumann Neighbour  $m$ 
3:      $payoff_{r,c} \leftarrow payoff_{r,c} + payoff_{Matrix}(r, c, neighbour)$  ▷ Matrix of  $T, R, P, S$ 
4:   end for
5: end procedure

1: procedure SUCCESS-DRIVEN MIGRATION( $r, c$ )
2:    $payoff_{max} \leftarrow payoff_{r,c}$  ▷ The Maximum Fictitious Payoff
3:    $r_{max}, c_{max} \leftarrow r, c$ 
4:   for  $neighbour \leftarrow neighbours$  do ▷ Empty Sites in Moore Neighbour  $M$ 
5:     if  $payoff_{neighbour} > payoff_{max}$  then ▷ Find the Best Neighbour
6:        $payoff_{max} \leftarrow payoff_{neighbour}$ 
7:        $r_{max}, c_{max} \leftarrow r_{neighbour}, c_{neighbour}$ 
8:     end if
9:   end for
10:   $strategy_{r_{max}, c_{max}} \leftarrow strategy_{r,c}$  ▷ Migrate to the Best Candidate
11:   $payoff_{r_{max}, c_{max}} \leftarrow payoff_{r,c}$ 
12:   $strategy_{r,c} \leftarrow empty$  ▷ Make Previous Site Clear
13:   $payoff_{r,c} \leftarrow 0$ 
14:  return  $r_{new}, c_{new}$  ▷ Return Migrated Site
15: end procedure

```

5.3 Reputation-Based Migration Model

In reputation-based migration model, individuals compute the reputation of neighbours and leaving probability based on it. Then, move to a random empty site and

imitate strategy as follows:

Algorithm 3 Reputation-Based Migration Model

```

1: procedure REPUTATION-BASED MIGRATION MODEL
2:   for  $iter \leftarrow 1$  to  $iter_{max}$  do
3:     for  $r, c \leftarrow grid$  do ▷ Random order
4:       Update Payoff( $r, c$ )
5:        $P \leftarrow \text{Compute Reputation}(r, c, \gamma, \alpha)$ 
6:        $r_{new}, c_{new} \leftarrow \text{Random Migration}(r, c, P)$ 
7:       Strategy Imitation( $r_{new}, c_{new}$ )
8:     end for
9:   end for
10: end procedure

```

Algorithm 4 Reputation-Based Migration

```

1: procedure COMPUTE REPUTATION( $r, c, \gamma, \alpha$ )
2:    $sum_{reputation} \leftarrow 0$  ▷ Sum of Neighbours Reputation
3:    $P \leftarrow 0$  ▷ The Probability of Leaving
4:   for  $neighbour \leftarrow neighbours$  do ▷ Empty Sites in Moore Neighbour  $M$ 
5:      $sum_{reputation} \leftarrow sum_{reputation} + reputation_{neighbour}^\gamma$ 
6:   end for
7:   if  $sum_{reputation} > 0$  then
8:      $P \leftarrow reputation_{r,c}^\gamma / sum_{reputation}$ 
9:   end if
10:   $reputation_{r,c} \leftarrow reputation_{r,c} \times \alpha$  ▷ Update Own reputation
11:  if  $strategy_{reputation} == cooperator$  then
12:     $reputation_{r,c} \leftarrow reputation_{r,c} + 1$ 
13:  end if
14:  return  $P$  ▷ Return the Probability of Leaving
15: end procedure

1: procedure RANDOM MIGRATION( $r, c, P$ )
2:   $neighbour \leftarrow rand(neighbours, P)$  ▷ Select an Empty Sites in Moore Neighbour  $M$ 
3:   $strategy_{r_{neighbour}, c_{neighbour}} \leftarrow strategy_{r,c}$  ▷ Migrate to the Neighbour
4:   $payoff_{r_{neighbour}, c_{neighbour}} \leftarrow payoff_{r,c}$ 
5:   $strategy_{r,c} \leftarrow empty$  ▷ Make Clear Previous Site
6:   $payoff_{r,c} \leftarrow 0$ 
7:  return  $r_{new}, c_{new}$  ▷ Return Migrated Site
8: end procedure

```

5.4 Our Model

Our model, which is the combination of success-driven migration and reputation-based migration model, is in the following:

Algorithm 5 Success-Driven and Reputation-Based Migration Model

```

1: procedure SUCCESS-DRIVEN AND REPUTATION-BASED MIGRATION MODEL
2:   for  $iter \leftarrow 1$  to  $iter_{max}$  do
3:     for  $r, c \leftarrow grid$  do ▷ Random order
4:       Update Payoff( $r, c$ )
5:        $P \leftarrow \mathbf{Compute Reputation}(r, c, \gamma, \alpha)$ 
6:        $r_{new}, c_{new} \leftarrow r, c$ 
7:       if  $P > rand()$  then ▷ with the Probability of Leaving
8:          $r_{new}, c_{new} \leftarrow \mathbf{Success-Driven Migration}(r, c)$ 
9:       end if
10:      Strategy Imitation( $r_{new}, c_{new}$ )
11:    end for
12:  end for
13: end procedure

```

5.5 Other models

To look at the migration patterns, we used three more models :

1. Success-Driven migration without imitation
2. Reputation-Based migration without imitation
3. Success-Driven and Reputation-Based migration without imitation

These models don't really represent real life situations but we used them compare how the different migrations work without the effect of imitation.

6 Simulation Results and Discussion

In this section, we present the results of our simulations.

6.1 Convergence

It is easy to see on the plots that the success-driven migration model, the model with imitation only and our model converge to a stable cooperator ratio in less than 5000 iterations. This was the case for all the different parameters α and γ that we tried. The success-driven migration model always converges faster than our model. This is because our model has a smaller rate of migration.

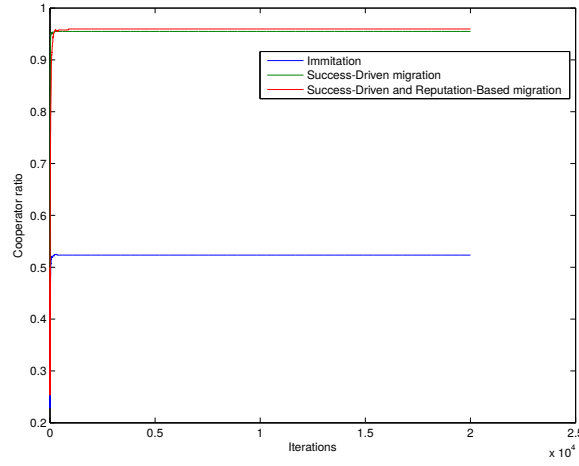
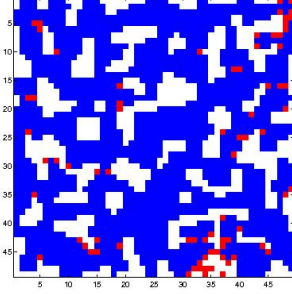
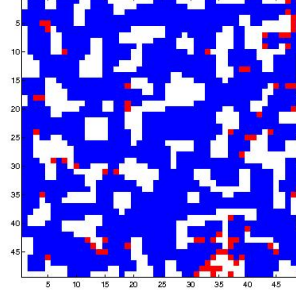


Figure 3: Cooperator ratios after a big number of iterations. We can see that the three models converge in about 2000 iterations. We used the parameters $\alpha = 0.5$ and $\gamma = 500$

The reputation-based migration model only reached a stable cooperator ratio after a number of iteration in the order of 30'000. This is due to the random nature of the migration. The player don't choose an empty place based on their interest to migrate but get assigned a random one. Our simulations didn't have results that could give us certainty that this stable cooperator ratio is always reached and that there won't be an instability after a greater number of iterations.

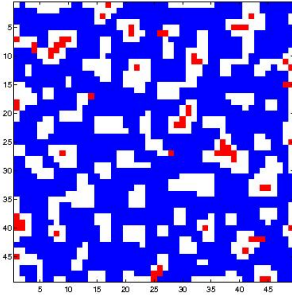


(a) Grid after 1000 iterations

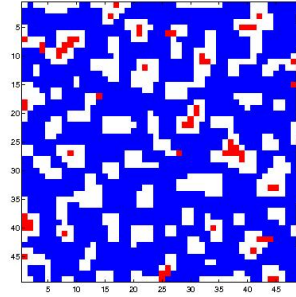


(b) Grid after 3000 iterations

Figure 4: Grids for the Success-Driven migration model after different number of iterations. We can see that it almost doesn't change and reached a stable state.



(a) Grid after 1000 iterations



(b) Grid after 3000 iterations

Figure 5: Grids for the Success-Driven and Reputation-Based migration model after different number of iterations. We can see that it almost doesn't change and reached a stable state. We used the parameters $\alpha = 0.5$ and $\gamma = 500$

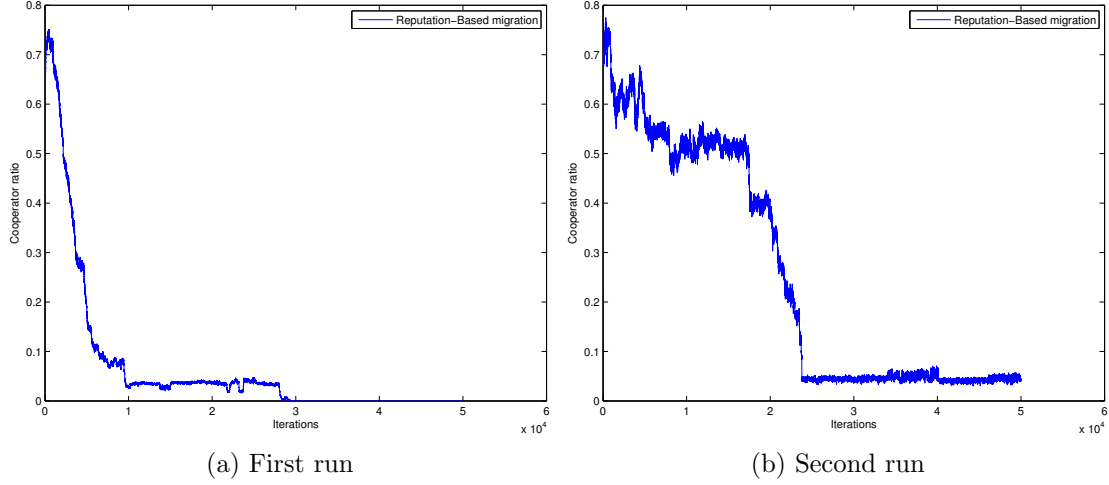


Figure 6: Cooperator ratios for Reputation-Based migration. We used the parameters $\alpha = 0.5$ and $\gamma = 500$

6.2 Cooperator ratios

To compare the cooperator ratios, we plotted the evolution of the ratios over the iterations for different parameters γ and α .

In figure 7, we see that, most of the time, our model has a slightly lower cooperator ratio than the success-driven migration model. We can see that for an α of 0.95, the cooperator ratio decreases a lot for γ over 200. This is because with these parameters, the probability of a player to leave decreases a lot (See figure 2, reputation gets big and the divisor gets bigger) and the model gets similar to the one with imitation only. For all other values, these parameters have no big impact on the cooperator ratios.

In figure 8, you can notice a drop in cooperator ratio at the beginning of the simulation for all the models which implement the imitation. This happened in all our simulations. The big rate of migration of the success-driven migrations compensate part of this drop.

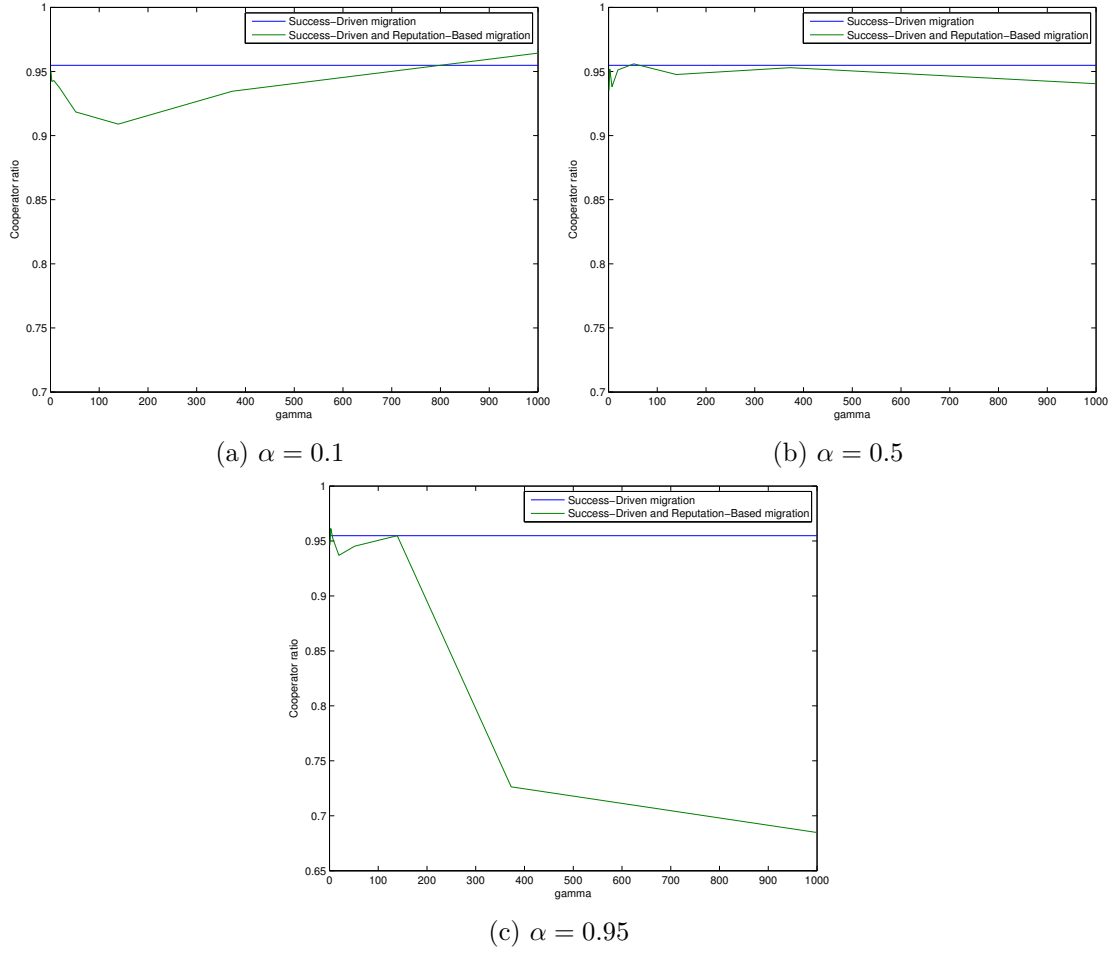


Figure 7: Cooperator ratios after 5000 iterations for different α and γ parameters

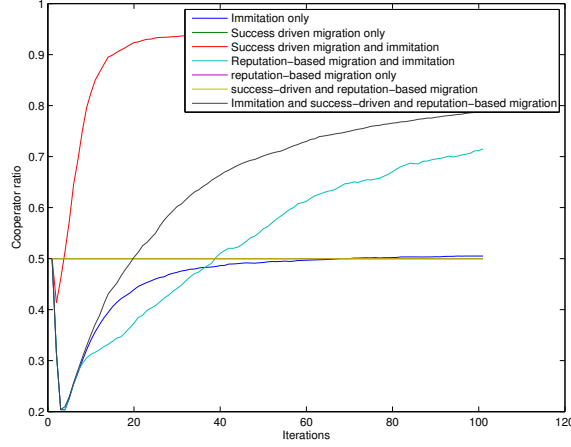


Figure 8: The first 100 iterations of a simulation. $\alpha = 0.5$ and $\gamma = 300$.

6.3 Patterns

In this part, we will look at how cooperators and defectors place themselves on the grid. We printed the grid for each model after a fixed number of iterations. We put a blue point where there was a cooperator, a red point where there was a defector and let a white space for each empty space.

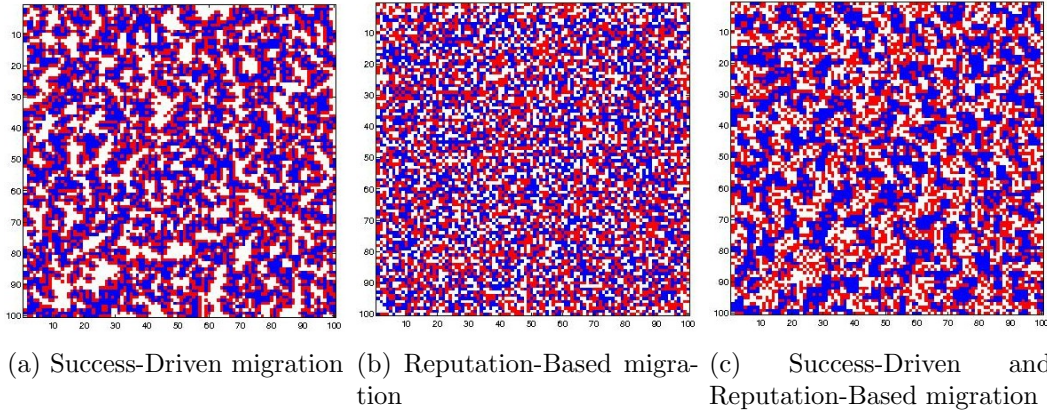


Figure 9: Grids after 200 iterations for the models without imitation. We used the parameters $\alpha = 0.5$ and $\gamma = 300$.

Figure 9 shows the patterns of the cooperators and defectors for the three models without imitation. In these models, the cooperator ratio stays fix at 50%. The first thing that we can see is that in our model (9c) and the success-driven migration

model (9a), the cooperators arrange themselves in clusters whereas they are randomly placed in the reputation-based migration model (9b). This randomness in the reputation-based migration model is again due to the players being placed randomly on the grid when they migrate. In the two other models, the players choose where they go based on their interest and thus the cooperators go near other cooperators because they give them the best payoff.

If we compare the success-driven migration model and our model, the main difference is in the patterns of the defectors. In the success-driven migration, the defectors are almost all placed around a cooperator cluster whereas in our model they are randomly placed on the grid. This difference can be explained by the fact that in our model, the players migrate only when their neighbours have a worse reputation than them. Since the defectors have a bad reputation, they generally don't move and stay at the random place they were assigned at the beginning. In the success-driven migration model, the defectors migrate and go around cooperators since they give them the best payoff.

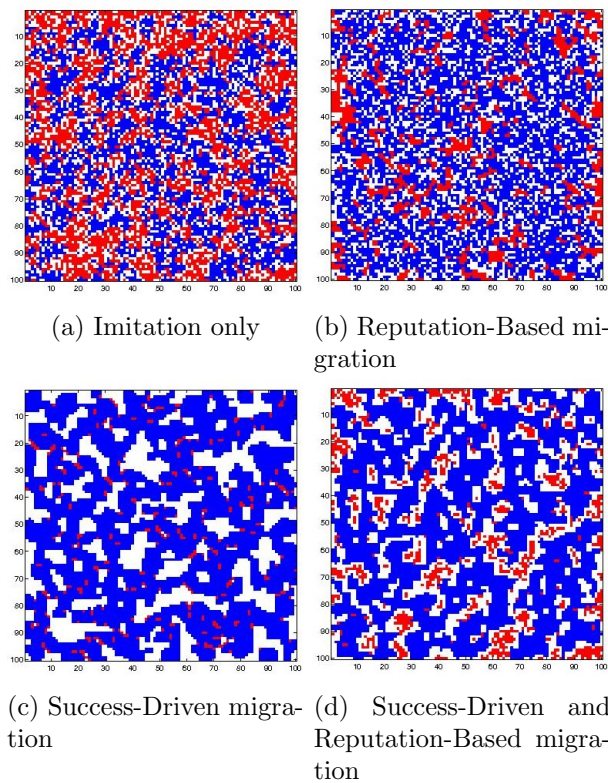


Figure 10: Grids after 200 iterations for the models with imitation. We used the parameters $\alpha = 0.5$ and $\gamma = 300$.

Figure 10 shows the patterns of cooperators and defectors in the models with imitation. We can see the same two effects previously analysed for the models without imitation, but in different forms. The randomness of placement in the reputation-based migration model (10b) induces a grid where the players are randomly placed on the grid and the empty space is randomly spread on the grid. In the success-driven migration model (10c) and our model (10d), the cooperators are organized in denser clusters because this is where they get the biggest payoff. This time, the difference between the success-driven migration model and our model is that in our model, the defectors are isolated. This is again because the defectors have little tendency to migrate in our model and the cooperator go away from them.

6.4 Discussion

All the three last points showed us that the combination of success-driven migration and reputation-based migration gives indeed in the results a combination of the two.

One of the drawback of the reputation-based migration model was its randomness and unpredictability. Our model gives stability to this reputation-based migration by adding the choice of the place to migrate. In terms of convergence, cooperator ratios and cooperator patterns, our model distinguish itself from the reputation-based migration model by getting the predictability of the success-driven migration model.

Our model doesn't get cooperator ratios as high as the success-driven migration model. However, it gets ratios higher than 90% with almost all the parameters α and γ we tested. This is higher than the reputation-based migration model.

Finally, our model manages to get this high ratio, almost as high as the one from the success-driven migration model, with a more natural way of migrating. Our model lets the player migrate based on their choice and we can see this in a slower rate of convergence. Even if the highest cooperator ratio needs more time to be reached, the cooperator still take over the defectors.

7 Summary and Outlook

We used MATLAB to implement the prisoner's dilemma game with imitation and with three different migration models :

1. Success-Driven migration
2. Reputation-Based migration
3. Success-Driven and Reputation-Based migration (our model)

We ran simulations of the game for the three different models and compared the results. The purpose was to find if our model, which is a more natural one based on choice instead of randomness, can have a cooperator ratio higher than or as high as the other models.

Our model was found to converge rapidly to a high ratio of cooperators. This convergence in thousands of iterations was similar to the one of the success-driven migration model but a little slower. It distinguished our model from the reputation-based migration model which didn't reach any stable state before tens of thousand of iteration.

The cooperator ratios our model converges to are almost as high as the one from the success-driven migration model but generally lower. They are still always above 90% except for some extreme parameters. It again distinguished our model from the reputation-based migration model which tendency was to have a ratio falling to 0%.

We didn't analyse our model under noisy conditions and an interesting next step would be to try and compare our model similarly to the work of Helbing and Yu in The outbreak of cooperation among success-driven individuals under noisy conditions [1]. Another interesting addition would be to try our model with different payoff parameters or temptation to defect.

8 References

1. Helbing, D., & Yu, W. (2009). The outbreak of cooperation among success-driven individuals under noisy conditions. *Proceedings of the National Academy of Sciences*, 106(10), 3680-3685.
2. Helbing, D., Yu, W., & Rauhut, H. (2011). Self-organization and emergence in social systems: Modeling the coevolution of social environments and cooperative behavior. *The Journal of Mathematical Sociology*, 35(1-3), 177-208.
3. Cong, R., Wu, B., Qiu, Y., & Wang, L. (2012). Evolution of cooperation driven by reputation-based migration. *PloS one*, 7(5), e35776.