Transaction on IoT and Cloud Computing 2(3) 1-10 2014

ISSN: 2331-4753 (Print)
ISSN: 2331-4761 (Online)

Communication scheme based on evolutionary spatial games in the cloud computing environment

Tautvydas Ragulskis[1], Vytautas Ostasevicius[1], Vytautas Pilkauskas[2], Minvydas Ragulskis[3]

[1] Institute of Mechatronics, Kaunas University of Technology, Kestucio 25A, Kaunas LT-44312, Lithuania

[2]Department of Applied Informatics, Kaunas University of Technology, Studentu 50, Kaunas LT-51368, Lithuania

[3]Corresponding author, email: minvydas.ragulskis@ktu.lt

Research Group for Mathematical and Numerical Analysis of Dynamical Systems

Kaunas University of Technology, Studentu 50-147, Kaunas LT-51368, Lithuania

Abstract

A new visual information communication scheme based on evolutionary spatial 2×2 games in the cloud computing environment is presented in this paper. Self-organizing patterns induced by complex interactions between competing individuals are exploited for hiding and transmitting secret digital dichotomous images. Evolutionary spatial game could service is used to distribute the pattern formation task in the cloud. Such approach enables to strengthen the security and to improve the efficiency of the proposed communication scheme.

Keywords: pattern formation, spatial game, cloud computing.

1 Introduction

The evolution of complex patterns in simple systems has attracted the attention of researchers for decades. One of the classical numerical examples illustrating a surprising variety of irregular spatiotemporal patterns comprises a simple reaction-diffusion model with finite amplitude perturbations [1]. A wide variety of two- and three-dimensional physical, chemical and biological systems displaying different morphologies of patterns is discussed in [2]. Small perturbations of initial states of the system play a central role in the initiation of pattern formation process [3-6].

It is well known that the interplay between competition and cooperation can originate the formation of spatial patterns. The model of cyclically competing species on a continuous space may result into spiral or plane-wave patterns [7]. Spatial predator-prey models can exhibit spatiotemporal pattern formation [8-11]. A model of six competing species results into a self-organizing pattern in [12]. The competition and cooperation is particularly well expressed in evolutionary spatial games (ESG). ESG models are widely discussed in the context of game theorybut they also provide a deep insight into physical properties of self-organizing systems and self-organizing patterns [13-16].

It has been demonstrated that self-organizing patterns can be efficiently exploited in hiding secret visual information. Self-organizing patterns induced by the Turing instability and produced by BeddingtonCDeAngelis-type predatorCprey model are successfully used in a secure steganographic communication algorithm in [17]. Communication scheme based on ESG is developed in [18]. The natural objective of further research is to extend such communication schemes to cloud computing platform. That would ensure two major advantages: security and efficiency. Breach of one in the cloud would not endanger the security of the whole scheme. Extensive parallelism in the cloud would result into efficiency of computations.

This paper is organized as follows: the ESG scheme is described in section 2; ESG cloud service is presented in section 3; random splitting of the cover image in the cloud is demonstrated in section 4; ESG Cloud Service Overlay is discussed in section 5; results of computational experiments are illustrated in section 6 and concluding remarks are given in the final section.

2 Preliminaries

As mentioned previously, we will use the communication scheme based on ESG [18] C the main attention in this paper will devoted to the adaptation of this scheme to cloud environment. But before discussing the basic principle of the communication scheme used in [18] we give the concise description of ESG:

(i) The cover image is a digital dichotomous image. Every pixel of the image is associated to the strategy of a single player; 1 represents defection and 0 represents cooperation.

(ii) Every single player plays eight 2×2 games (the prisoners dilemma game) with all his neighbors. The outcome of a single static 2×2 game is represented by the payoff matrix: Payoffs for Player 1 (P_1) and Player 2 (P_2) are shown in bold and normal fonts

	P_2 cooperates	P_2 defects
P_1 cooperates	R, R	S, T
P_1 defects	T, S	P, P

accordingly. R is the "reward" both players receive if they both cooperate; P is the "punishment" payoff both players receive if they both defect. If P_1 defects while P_2 cooperates, then P_1 receives the "temptation" payoff T while P_2 receives the "suckers" payoff S. Similarly, if P_1 cooperates while P_2 defects, then P_1 receives S while P_2 receives T [19].

(iii) The sum of eight payoffs (8 games with adjacent neighbors) is associated to every player.

(iv) Every single player looks around his neighborhood (8 adjacent players + himself) and selects the player (from this neighborhood) with the highest payoff. The strategy of the most successful player (1 or 0) is adopted to current player.

(v) Iterations in space and time result into ESG model.

Of course, not all parameter values R, S, T, P result into the pattern formation process; we use the values selected in [18]: $R = 3$; $S = 2$; $T = 4$; $P = 0$.

The communication algorithm between Bob (the sender) and Alice (the receiver) can be described as follows. Bob generates an initial population of players (a random matrix) and modifies this dichotomous picture by inverting pixels which are located in the regions occupied by the secret image. Then Bob runs the iterative ESG model of the grid for a predetermined number of steps and saves the image of the pattern. Bob sends this dichotomous pattern via an open communication channel to Alice. Bob also sends 2 public keys, the seed value of the random number generator used to generate the initial conditions and the number of forward iterations for the ESG model.

Alice, upon receiving the seed value, generates the unperturbed copy of initial population (the size of the grid is given by the dimensions of the received image from Bob). Then she runs the iterative ESG algorithm on the grid (starting from the unperturbed initial population) for the predefined number of forward steps. Finally, Alice performs XOR subtraction between her image and the image received from Bob. The difference image leaks the shape of the secret image encoded into the initial population by Bob [18].

3 Communication scheme based on ESG cloud service

As mentioned previously, the basic structure of the ESG algorithm will be preserved as used in [18]. The major difference is in the communication between Bob and Alice (Fig. 1). Bob does not send the pattern which had evolved from perturbed initial conditions, he just delivers the perturbed cover image to ECG Cloud Service.

Bob (gray background) and Alice (dotted area) start from the same initial conditions (Fig. 1), but Bob does perturb these initial conditions by inverting pixels in the zones corresponding to the secret image (bottom left) which is known only to Bob. Bob sends the perturbed initial conditions to the ESG Cloud Service which does comprise s different nodes; the digital image of perturbed initial conditions is randomly split among nodes. ESG Cloud Service evolves the pattern and sends partitioned images to Alice. Alice does not only assemble the pattern image received from the ESG Cloud Service, she also evolves the pattern from the unperturbed initial conditions. The XOR difference between two images yields the secret image.

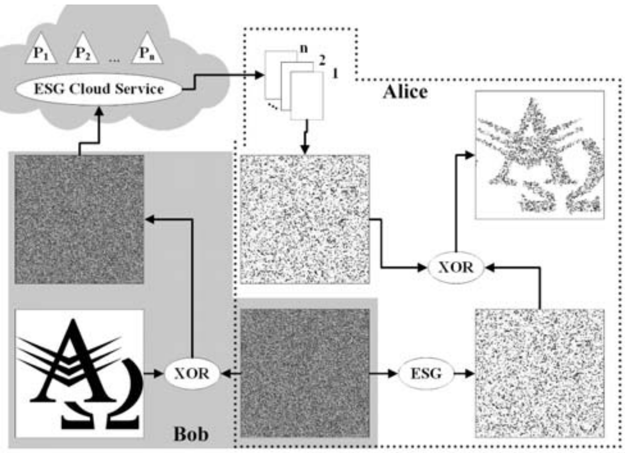

Fig. 1. Bob (the gray background) and Alice (the dotted area) start from the same initial conditions (bottom center), but Bob does perturb these initial conditions by inverting pixels in the zones corresponding to the secret image (bottom left). Alice does not know the secret image—she simply uses ESG algorithm to produce the pattern from unperturbed initial conditions. XOR difference between Alice's and Bob's patterns leaks the secret (middle right).

4 Random splitting of the cover image in the cloud

The random splitting procedure among s different nodes in the cloud is illustrated in Fig. 2; the pseudo-code is listed in **procedure** random_splitting. There are two important aspects associated with this splitting rule. The first one is the requirement that the assembly procedure would be able to reproduce the splitting sequence. In other words, we do require that the random number generator would be reproducible from a pre-selected seed value-similar to the generation of random initial conditions for the ESG Cloud Service. In other words, the encoding algorithm requires two seed values, one for the generation of random initial conditions; the other one for splitting these conditions into s nodes. Both seed values are internal keys are must be transmitted by Bob to Alice before the decryption procedure can be commenced.

The other aspect of the random splitting procedure is associated with the fact that every pixel must be used to in a 2×2 evolutionary game with its all 8 surrounding neighbor pixels (Fig. 2). In other words, every node in the ESG Cloud Service must not only know the address of the current pixel in the global matrix, but also needs to know the numbers of nodes and addresses of all of 8 of its adjacent neighbors. But since the pixels are distributed among the nodes in a random rule, adjacent pixels to the current pixel can be located in different nodes. Such situation is illustrated in Fig. 2, the current pixel is located in the node s_h; but its fifth neighbor, in the node s_k. All addressing rules must take this allocation into account; the details of the splitting are represented in the pseudo-code below:

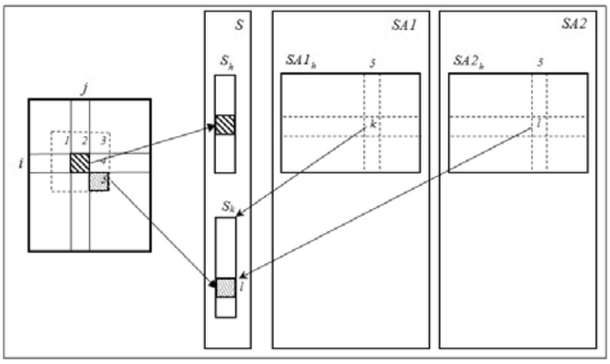

Fig. 2. The random splitting procedure among s different nodes in the cloud.

```
procedure random_splitting(X[n,m],S[s][ns],SA1[s][ns,8],SA1[s][ns,8])
  assign memory to array A1 of size [n,m];
  assign memory to array A2 of size [n,m];
  assign memory to array C of size [s];
  initialize random generator using the specified seed value;
  for i = 1 to n
    for j = 1 to m {
      repeat k = random from 1 to s while (C[k] > ns);
      A1[i, j] = k; A2[i, j] = C[k];
      S[k][C[k]] = (X[i, j]);
      C[k]++;
    }
  for i = 1 to n
    for j = 1 to m
      for k = 1 to 8 {
        calculate ii,jj for immediate neighbors;
        SA1[A1[i, j]][A2[i, j], k] = A1[ii, jj];
        SA2[A1[i, j]][A2[i, j], k] = A2[ii, jj];
      }
end procedure
```

The scattered images of the developed pattern produced by individual nodes in the ESG Cloud Service must be merged into one pattern (after separate images are acquired by the receiver). The random number generator must use the same seed value which was used during the random splitting procedure; otherwise the pseudo-code is straightforward:

```
procedure merge(S[s][ns], Y[n,m])
  assign memory to array C of size [s];
  initialize random generator using the specified seed value;
  for i = 1 to n
    for j = 1 to m {
      repeat k = random from 1 to s while (C[k] > ns);
      Y[i, j] = S[k][C[k]];
      C[k]++;
    }
end procedure
```

5 ESG Cloud Service Overlay

ESG Cloud Service consist of the service overlay ant the peer-to-peer network (Fig. 3). The procedure random_splitting() is realized in the service overlay.

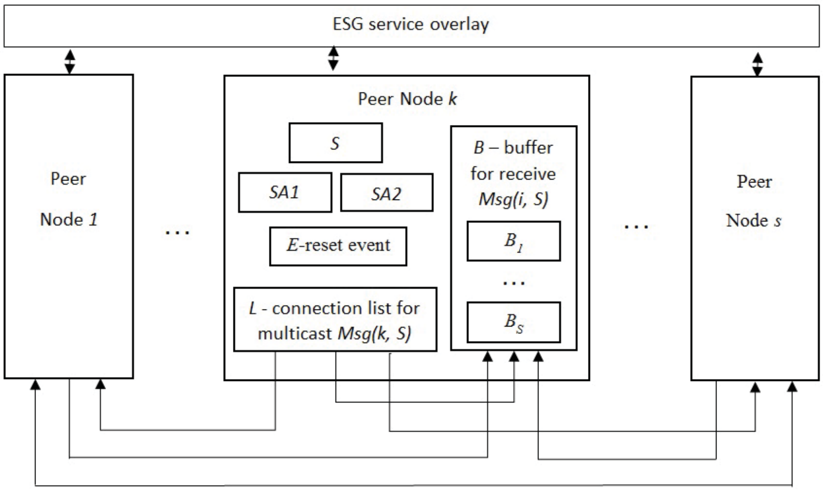

Fig. 3. The schematic diagram of ESG Cloud Service.

Two threads are realized in the peer node:

- procedure compute_ESG();
- procedure receive_Msg();

It is important to note that a reset event E is used for synchronization this treads. Procedure computeESG() gets S, SA1, SA2 data from ESG service overlay and computes ESG algorithm steps which realize the formation of the pattern:

```
procedure compute_ESG()
  initialize event E to false;
  get k,S,SA1,SA2 from ESG service overlay for curent peer node;
  for p = 1 to number_of_steps {
    multicast Msg(k,S);
    wait event E;
    reset event E to false;
    compute single ESG step from B to S
  }
  return S to service overlay;
end procedure
```

On receiving message every peer has to call the procedure receive_Msg(i,S). Note that the arguments of this procedure are the current state S of the peer node i:

```
procedure receive_Msg(i, S[ns])
  for j = 1 to ns
    B[i, j] = S[j];
  if buffer B full {
    set event E to true;
    clear buffer B;
    }
end procedure
```

The single ESG step does not differ from the technique implemented on a single server in [18]. It can be illustrated in such a schematic description. Let $M(i,j)$ is a player located at ith row and jth column of a rectangular grid with periodic boundaries; $i = 1, 2, \cdots, n$; $j = 1, 2, \cdots, m$; where n and m define the size of the cover image with periodic boundaries. The entries of the matrix M are binary digits, 1 representing defection and 0 representing cooperation of the current player. Then the payoff $U(i,j)$ for the player $M(i,j)$ reads:

$$U(i,j) = \sum_{(k,l)\in I(i,j)} M(i,j)(M(k,l)P + (1 - M(k,l))T) + (1 - M(i,j))(M(k,l)S + (1 - M(k,l))R) \quad (1)$$

where $I(i,j)$ is the set of indices of eight closest neighbors of the player $M(i,j)$; the values of parameters P, T, S and R are described in Section 2. The payoff matrix U is used to update the strategy of players in the next time step:

$$\hat{M}(i,j) = M(r,s) \quad (2)$$

where $\hat{M}(i,j)$ is the strategy of the player $M(i,j)$ in the next time step; (r,s) are the coordinates of the neighbor who has previously received the highest payoff:

$$(r,s) = \arg \max_{(k,l)\in I(i,j)\cup\{(i,j)\}} U(k,l) \quad (3)$$

The payoff matrix U is recalculated as soon as strategies for all players are updated.

6 Computational experiments

We will use ESG Cloud Service comprising 10 nodes to illustrate the functionality of the proposed communication scheme. The size of the cover image is selected to hold 200 $\times$ 200 players; initial player strategies are generated by a pseudo-random number generator (the chaotic logistic map) with a preselected seed value (Fig. 4). The secret image (known only to Bob at the beginning of the computational experiment) is also shown in Fig. 4.

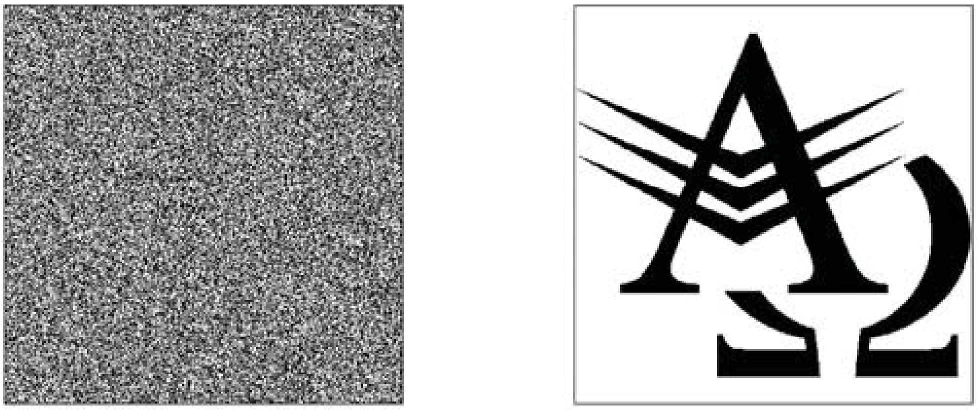

Fig. 4. The unperturbed initial random conditions (200 x 200 players) and the secret image.

The perturbed initial conditions are illustrated in upper left of Fig. 5. The perturbed initial conditions are split into 10 nodes (random_splitting procedure is used for that purpose). Compute_ESG procedure is used to evolve the pattern in the cloud; 10 different randomly split patterns are illustrated at the bottom of Fig. 5. Note that an eavesdropper having access to a single ESG pattern in one of the nodes has no chance to break the secret, the array of pixels in any of the nodes is randomly distributed in the cover image of the pattern evolved from the perturbed initial conditions. The evolved pattern can be reconstructed only if the receiver known the seed value of the random splitting procedure.

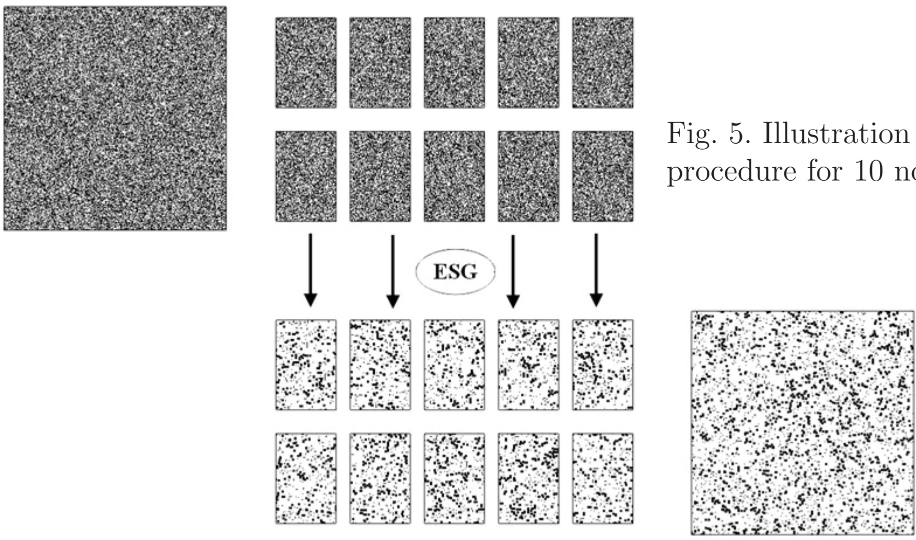

Fig. 5. Illustration of the ESG procedure for 10 nodes in the cloud.

Finally, the receiver has to run the ESG algorithm for the non-perturbed initial conditions, the resulting pattern is shown in Fig. 6. Note that the number of time forward steps in the ESG evolution must coincide exactly with the number of steps used for the perturbed initial conditions. The XOR difference between two patterns results into the shape corresponding to the initial secret image (Fig. 4).

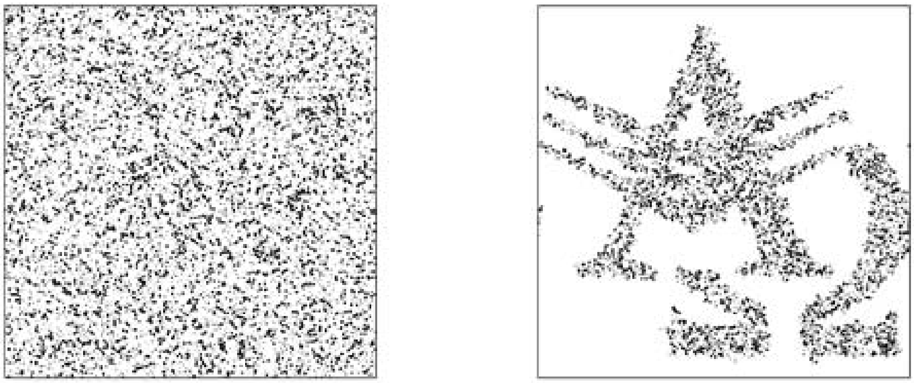

Fig. 6. The ESG pattern produced by the non-perturbed initial conditions and the reconstructed secret image.

7 Concluding remarks

Communication scheme based on evolutionary spatial 2 × 2 games in the cloud computing environment is presented in this paper. We do not only describe the basic structure of the communication algorithm, but give the detailed description of the implementation. ESG Cloud Service, the architecture of peer nodes, service overlay is discussed in details.

The extension of the communication scheme to the cloud environment yields two major advantages (compared to conventional non-cloud communication schemes). The first advantage is the increased security of the scheme. Even if an eavesdropper would be able to break a single peer, he would not be able to reconstruct the secret. Random splitting between nodes prevents an ability to reconstruct even a slight part of the secret.

The second advantage is the efficiency of the implementation. Conventional evolutionary pattern based single node communication schemes work well for static images. But the initial conditions (perturbed and non-perturbed) must be allowed to evolve into patterns. That could be a time-consuming task for larger cover images. The outsourcing of this task to a cloud changes the situation completely. Massive parallelism in the cloud could increase the efficiency of the scheme to a level when one could consider a real time video secret communication scheme. That is a definite object of the future research.

Acknowledgment

Financial support from the Lithuanian Science Council under project No. MIP-100/12 is acknowledged.

References

[1] J. E. Pearson. Complex patterns in a simple system. Science 261(5118), 189 (1993).

[2] M. Seul, D. Andelman. Domain shapes and patterns: the phenomenology of modulated phases. Science 267(5197), 476 (1995).

[3] K. J. Lee, W. D. McCormick, Q. Ouyang, H. L. Swinney. Pattern formation by interacting chemical fronts. Science 261(5118), 192 (1993).

[4] S. Kondo, T. Miura. Reaction-diffusion model as a framework for understanding biological pattern formation. Science 329(5999), 1616 (2010).

[5] A. Yochelis, M. Sheintuch. Principal bifurcations and symmetries in the emergence of reaction-diffusion-advection patterns on finite domains. Phys. Rev. E 80, 056201 (2009).

[6] S. Sen, P. Ghosh, S. S. Riaz, D. S. Ray. Time-delay-induced instabilities in reaction-diffusion systems. Phys. Rev. E 80, 046212 (2009).

[7] X. Ni,W.-X.Wang, Y.-C. Lai, C. Grebogi. Evolutionary game theory and the emergence of spatio-temporal order in ecological systems. Phys. Rev. E 82, 066211 (2010).

[8] W. Wang, Y. Lin, L. Zhang, F. Rao, Y. Tan. Pattern formation in a predatorCprey model with spatial effect. Comm. Nonlin. Sci. Numer. Simulat. 16, 2006 (2011).

[9] W. Wang, L. Zhang, H. Wang, Z. Li. Pattern formation of a predator-prey system with Ivlev-type functional response. Ecol. Model. 221(2), 131 (2010).

[10] M. R. Garvie, C. Trenchea. Spatiotemporal dynamics of two generic predator-prey models J. Biol. Dyn. 4(6), 559 (2010).

[11] B. Dubey, N. Kumari, R. K. Upadhyay. Spatiotemporal pattern formation in a diffusive predator-prey system: an analytical approach. J. Appl. Math. Comput. 31, 413 (2009).

[12] G. Szabo, A. Szolnoki, I. Borsos. Self-organizing patterns maintained by competing associations in a six-species predator-prey model. Phys. Rev. E 77, 041919 (2008).

[13] M. Perc, A. Szolnoki. Coevolutionary games C a mini review. Biosystems 99 (2) 109-125 (2010).

[14] G. Szabo, G. Fath. Evolutionary games on graphs. Phys. Rep. 446 (4) 97-216 (2007).

[15] M. Perc, J. Gomez-Gardenes, A. Szolnoki, L.M. Floria, Y. Moreno. Evolutionary dynamics of group interactions on structured populations: a review. J. R. Soc. Interface 10 (80) (2013).

[16] Ping-Ping Li, Jianhong Ke, Zhenquan Lin, P.M. Hui. Cooperative behavior in evolutionary snowdrift games with the unconditional imitation rule on regular lattices. Phys. Rev. E 85 (2) 021111 (2012).

[17] L. Saunoriene, M. Ragulskis. Secure steganographic communication algorithm based on self-organizing patterns. Phys. Rev. E 84 056213 (2011).

[18] P. Ziaukas, T. Ragulskis, M. Ragulskis. Communication scheme based on evolutionary spatial 2×2 games. Physica A 403 177-188 (2014).

[19] H. Fort, E. Sicardi, Evolutionary Markovian strategies in 2×2 spatial games. Physica A 375 323-335 (2007).