A Rough Set Version of the Go Game on Physarum Machines

Andrew Schumann
University of Information Technology and Management
Sucharskiego Str. 2
35-225 Rzeszów, Poland
andrew.schumann@gmail.com

Krzysztof Pancerz
University of Information Technology and Management
Sucharskiego Str. 2
35-225 Rzeszów, Poland
Chair of Computer Science, Faculty of Mathematics and Natural Sciences
University of Rzeszów
Prof. S. Pigonia Str. 1
35-310 Rzeszów, Poland
kkpancerz@gmail.com

ABSTRACT

We make use of a *Physarum* machine that is a biological computing device implemented in the plasmodium of *Physarum polycephalum* and/or *Badhamia utricularis* which are one-cell organisms able to build complex networks for solving different computational tasks. The plasmodium behavior can model an ancient Chinese game called Go. In the paper, we describe implementation of the Go game on the *Physarum* machines. A special version of the game is presented, where payoffs are assessed by means of the measure defined on the basis of rough set theory. Theoretical foundations given in the paper are supplemented with description of a specialized software tool developed, among others, for simulation of the described game.

Categories and Subject Descriptors

F.1.2 [**Modes of Computation**]: Parallelism and concurrency; I.2.1 [**Applications and Expert Systems**]: Games; F.4.1 [**Mathematical Logic**]: Set theory

General Terms

Theory

Keywords

rough sets, antagonistic game, Physarum machines

1. INTRODUCTION

Go is a game, originated in ancient China, in which two people play with a Go board and Go stones (cf. [5]). In general, the two players alternately place black and white stones, on the vacant intersections of a board with a 19×19 grid of lines, to surrounding territory. Whoever has more territory at the end of the game is the winner. Vertically and horizontally adjacent stones of the same color form a group. One of the basic principles of Go is the fact that stones must have at least one *liberty* to remain on the board. A liberty of a given stone is a vacant intersection adjacent to it. If a stone has at least one liberty, then the next stone of a given player can be placed on it to extend his/her group.

As it was shown in [12], the Go game can be simulated by means of a *Physarum* machine. A *Physarum* machine is a programmable amorphous biological computing device experimentally implemented in the plasmodium of *Physarum polycephalum*, also called true slime mould [1]. *Physarum polycephalum* is a single cell organism belonging to the species of order *Physarales*. General assumptions for games implemented on *Physarum* machines were presented in [16] and [17]. In our case, we assume that *Physarum* machines are experimentally implemented in the plasmodium of *Physarum polycephalum*, as well as in the plasmodium of *Badhamia utricularis* (cf. [7], [17]). *Badhamia utricularis* is another species of order *Physarales*. The plasmodium of *Physarum polycephalum* or *Badhamia utricularis*, spread by networks, can be programmable by adding and removing attractants and repellents. Two syllogistic systems implemented as Go games were considered in [12], namely, the Aristotelian syllogistic as well as the performative syllogistic. In the first case, the locations of black and white stones are understood as locations of attractants and repellents, respectively. In the second case, the locations of black stones are understood as locations of attractants occupied by plasmodia of *Physarum polycephalum* and the locations of white stones are understood as locations of attractants occupied by plasmodia of *Badhamia utricularis*. The Aristotelian syllogistic version of the Go game is a coalition game. The performative syllogistic version of the Go game is an antagonistic game.

In the paper, we will consider the second case (i.e., an antagonistic game implemented in plasmodia of *Physarum polycephalum* and *Badhamia utricularis*). For this case, we propose a new version of the Go game. In the presented approach, payoffs are assessed by means of the measure de-

.

BICT 2015, December 03-05, New York City, United States
Copyright © 2016 ICST
DOI 10.4108/eai.3-12-2015.2262488

fined on the basis of rough set theory. Rough sets are a tool to deal with rough (ambiguous, imprecise) concepts in the universe of discourse (cf. [10]). In our previous research, we used rough sets to describe behavior of *Physarum* machines (see [8] and [14]). In this case *Physarum* machines were modeled by means of transition systems or timed transition systems and rough set models were more abstract models built over transition systems. As it was shown, in rough set descriptions of *Physarum* machines, both a standard definition of rough sets proposed by Z. Pawlak [10] and the Variable Precision Rough Set Model (VPRSM) can be used. The VPRSM approach is a more relaxed and generalized rough set approach, proposed by W. Ziarko in [18]. Basic definitions, notions and notation concerning rough set theory as well as the VPRSM approach are recalled in Section 2. A rough set based approach was also used by us to describe strategy games implemented on *Physarum* machines [7]. The strategies of such games were approximated on the basis of a rough set model, describing behavior of the *Physarum* machine, created according to the VPRSM approach.

In the paper, another rough set based approach to assessing payoffs of games implemented on *Physarum* machines is proposed in Section 3. This time, for the Go game, we do not use transition system models describing behavior of *Physarum* machines, but our attention is focused on the adjacent surroundings of stones. Rough sets are used to determine how (i.e., exactly or roughly) these surroundings approximate sets of intersections occupied by either black or white stones (i.e., occupied by either plasmodia of *Physarum polycephalum* or plasmodia of *Badhamia utricularis*).

To support research on programming *Physarum* machines and simulating *Physarum* games, we are developing a specialized software tool, called the *Physarum* software system, shortly *PhysarumSoft* (see [13]). The approach presented in this paper is also implemented in *PhysarumSoft*. A short description of the *Physarum* game module is given in Section 4.

2. THE RUDIMENTS OF ROUGH SETS

In this section, we recall necessary definitions, notions and notation concerning rough sets.

The idea of rough sets (see [10]) consists of the approximation of a given set by a pair of sets, called the lower and the upper approximation of this set. Some sets cannot be exactly defined. If a given set X is not exactly defined, then we employ two exact sets (the lower and the upper approximation of X) that define X roughly (approximately).

Let $U \neq \emptyset$ be a finite set of objects we are interested in. U is called the universe. Any subset $X \subseteq U$ of the universe is called a concept in U. Let R be any equivalence relation over U. We denote an equivalence class of any $u \in U$ by $[u]_R$. With each subset $X \subseteq U$ and any equivalence relation R over U, we associate two subsets:

- $R_*(X) = \{u \in U : [u]_R \subseteq X\}$,
- $R^*(X) = \{u \in U : [u]_R \cap X \neq \emptyset\}$,

called the R-lower and R-upper approximation of X, respectively. A set $BN_R(X) = R^*(X) - R_*(X)$ is called the R-boundary region of X. If $BN_R(X) = \emptyset$, then X is sharp (exact) with respect to R. Otherwise, X is rough (inexact).

The definitions given earlier are based on the standard definition of set inclusion. Let U be the universe and $A, B \subseteq U$. The standard set inclusion is defined as

$$A \subseteq B \text{ if and only if } \underset{u \in A}{\forall} u \in B.$$

In some situations, the application of this definition seems to be too restrictive and rigorous. W. Ziarko proposed in [18] some relaxation of the original rough set approach. His proposition was called the Variable Precision Rough Set Model (VPRSM). The VPRSM approach is based on the notion of the majority set inclusion. Let U be the universe, $A, B \subseteq U$, and $0 \leq \beta < 0.5$. The majority set inclusion is defined as

$$A \overset{\beta}{\subseteq} B \text{ if and only if } 1 - \frac{card(A \cap B)}{card(A)} \leq \beta,$$

where $card$ denotes the cardinality of the set. $A \overset{\beta}{\subseteq} B$ means that the specified majority of elements belonging to A belongs also to B. One can see that if $\beta = 0$, then the majority set inclusion becomes the standard set inclusion.

By replacing the standard set inclusion with the majority set inclusion in definitions of approximations, we obtain the following two subsets:

- $R_*^\beta(X) = \{u \in U : [u]_R \overset{\beta}{\subseteq} X\}$,
- $R^{*\beta}(X) = \{u \in U : \frac{card([u]_R \cap X)}{card([u]_R)} > \beta\}$,

called the R_β-lower and R_β-upper approximation of X, respectively.

3. ROUGH SET BASED ASSESSMENT OF PAYOFFS

In this section, we present basic principles of the rough set version of the Go game implemented on the *Physarum* machine. Let a board for the Go game, with a 19×19 grid of lines, be in use. The set of all intersections of the grid is denoted by I. At the beginning, the fixed numbers of original points of both the plasmodia of *Physarum polycephalum* and the plasmodia of *Badhamia utricularis* are randomly deployed on intersections. An example of the initial configuration of the Go game is shown in Figure 1. In this case, three original points of the plasmodia of *Physarum polycephalum*, Ph_1, Ph_2, and Ph_3, understood as black stones, as well as three original points of the plasmodia of *Badhamia utricularis*, Ba_1, Ba_2, and Ba_3, understood as white stones, are deployed on intersections.

During the game, the two players alternately place attractants on the vacant intersections of the board. The first player plays for the *Physarum polycephalum* plasmodia, the second one for the *Badhamia utricularis* plasmodia. The plasmodia look for attractants, propagate protoplasmic veins towards them, feed on them and go on. In a real-life implementation of *Physarum* machines, attractants are sources of nutrients or pheromones, on which the plasmodium feeds (see [1]). The attractants occupied by plasmodia of *Physarum polycephalum* are treated as black stones whereas the attractants occupied by plasmodia of *Badhamia utricularis*, as white stones.

Remark 1. *We assume that the state of the Go game (i.e., the state of the Physarum machine) is observed only*

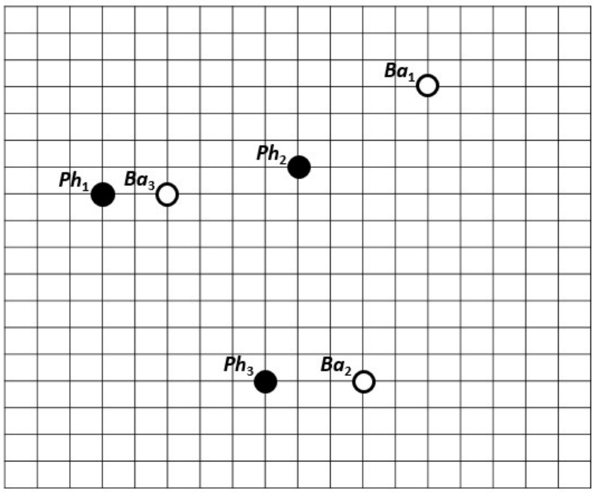


Figure 1: An example of the initial configuration of the Go game implemented on the *Physarum* machine

at discrete time instants, i.e., after each move in the game. Therefore, whenever time instant t is used, it means that $t = 0, 1, 2, \ldots$.

For our version of Go, we assume the following formal structure of the *Physarum* machine. The *Physarum* machine is a triple:

$$\mathcal{PM} = (P, B, A),$$

where:

- $P = \{Ph_1, Ph_2, \ldots, Ph_k\}$ is a set of original points of the plasmodia of *Physarum polycephalum.*
- $B = \{Ba_1, Ba_2, \ldots, Ba_l\}$ is a set of original points of the plasmodia of *Badhamia utricularis.*
- $A = \{A^t\}_{t=0,1,2,\ldots}$ is the family of the sets of attractants, where $A^t = \{A_1^t, A_2^t, \ldots, A_{r_t}^t\}$ is the set of all attractants present at time instant t in $\mathcal{PM}$.

One can see that the structure of the *Physarum* machine $\mathcal{PM}$ is changing in time (new attractants are added by the players). Positions of original points of the plasmodia as well as attractants are considered in the two-dimensional space of intersections. Hence, each intersection $i \in I$ is identified by two coordinates x and y. This fact will be denoted by $i(x, y)$. For each intersection $i(x, y)$, we can distinguish its adjacent surroundings:

$$\begin{aligned} Surr(i) &= \{i'(x', y') \in I : \\ &(x' = x - 1 \lor x' = x + 1) \land (x' \geq 1) \land (x' \leq 19) \\ &\land \\ &(y' = y - 1 \lor y' = y + 1) \land (y' \geq 1) \land (y' \leq 19)\}. \end{aligned}$$

It is worth noting that liberties of stones placed at intersections are defined only in adjacent surroundings of stones.

After each move (i.e., placement of a new attractant on the board by one of the players), the following rules of behavior of the plasmodia are applied:

1. As soon as the attractant is placed on an intersection and the adjacent surroundings contain the plasmodium of either *Physarum polycephalum* or *Badhamia utricularis*, this new attractant is occupied by the plasmodium.
2. If there exist more than one intersection occupied by both the plasmodia of *Physarum polycephalum* and the plasmodia of *Badhamia utricularis* in the adjacent surroundings of the intersection where the attractant was placed, then only the plasmodia of one type (randomly selected) are attracted. The plasmodium of *Physarum polycephalum* and the plasmodium of *Badhamia utricularis* cannot occupy the same attractants.

In case of Rule 2, the fusion of the plasmodia of one type can hold. Moreover, Rule 2 causes that, in some situations, given moves are taken under the players' own risk. Let us consider an illustrative configuration of the Go game shown in Figure 2. There are two attractants occupied by the plasmodia of *Physarum polycephalum* (black stones) and one attractant occupied by the plasmodia of *Badhamia utricularis* (white stone) in the adjacent surroundings of the intersection marked with $\times$. If the first player places a new attractant $A_\times$ at intersection $\times$, two situations, mutually exclusive, will be possible. In the first situation, two plasmodia of *Physarum polycephalum* will be attracted by $A_\times$ and fused on it. In this case, a new black stone will appear. In the second situation, the plasmodium of *Badhamia utricularis* will be attracted by $A_\times$. In this case, a new white stone will appear. It means that the first player's move may work to the advantage of the opponent.

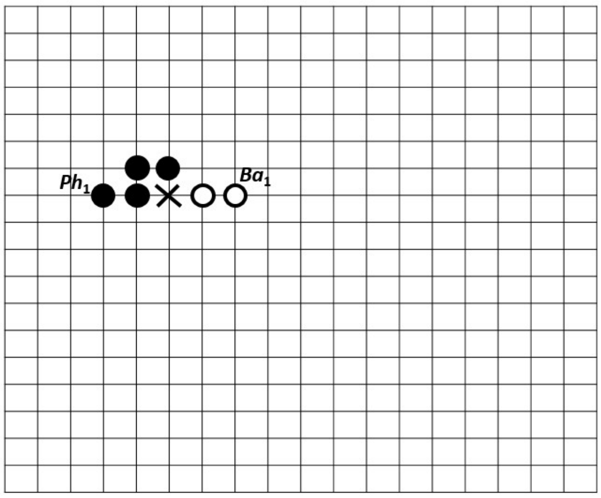


Figure 2: An illustrative configuration of the Go game, where a move is taken under the player's own risk

Formally, during the game, at given time instant t, we can distinguish three kinds of intersections in the set I_t of all intersections:

- $I_t^\emptyset$ - a set of all vacant intersections at t.
- $I_t^\bullet$ - a set of all intersections occupied by plasmodia of *Physarum polycephalum* at t (black stones).
- I_t° - a set of all intersections occupied by plasmodia of *Badhamia utricularis*) at t (white stones).

One can see that $I_t = I_t^{\emptyset} \cup I_t^{\bullet} \cup I_t^{\circ}$, where $I_t^{\emptyset}$, $I_t^{\bullet}$, and I_t° are pairwise disjoint.

A set I_t^{π} of all intersections occupied by given plasmodia π (either the plasmodia of *Physarum polycephalum* or the plasmodia of *Badhamia utricularis*) at given time instant t can be approximated (according to rough set definition) by surroundings of intersections occupied by these plasmodia. Such approximation will be called surroundings approximation.

The lower surroundings approximation $Surr_*(I_t^{\pi})$ of I_t^{π} is given by configuration of the Go game.

$$Surr_*(I_t^{\pi}) = \{i \in I_t^{\pi} : Surr(i) \neq \emptyset \wedge Surr(i) \subseteq I_t^{\pi}\},$$

where π is either $\bullet$ or $\circ$. The lower surroundings approximation consists of all intersections, occupied by plasmodia π, whose all of not vacant adjacent intersections are also occupied by π. Each intersection $i \in I$ such that $i \in Surr_*(I_t^{\pi})$ is called a full generator of the payoff of the player playing for the plasmodia π.

The upper surroundings approximation $Surr^*(I_t^{\pi})$ of I_t^{π} is given by

$$Surr^*(I_t^{\pi}) = \{i \in I_t^{\pi} : Surr(i) \cap I_t^{\pi} \neq \emptyset\},$$

where π is either $\bullet$ or $\circ$. The upper surroundings approximation consists of all intersections, occupied by plasmodia π, whose adjacent surroundings cover at least one intersection also occupied by π.

The set $BN_{Surr}(I_t^{\pi}) = Surr^*(I_t^{\pi}) - Surr_*(I_t^{\pi})$ is referred to as the boundary region of surroundings approximation of I_t^{π} at time instant t. Each intersection $i \in I$ such that $i \in BN_{Surr}(I_t^{\pi})$ is called a partial generator of the payoff of the player playing for the plasmodia π.

By replacing the standard set inclusion with the majority set inclusion in the definition of the lower surroundings approximation (according to the VPRSM approach recalled in Section 2), we obtain the β-lower surroundings approximation:

$$Surr_*^{\beta}(I_t^{\pi}) = \{i \in I_t^{\pi} : Surr(i) \neq \emptyset \wedge Surr(i) \overset{\beta}{\subseteq} I_t^{\pi}\},$$

Each intersection $i \in I$ such that $i \in Surr_*^{\beta}(I_t^{\pi})$ is called a full quasi-generator of the payoff of the player playing for the plasmodia π.

On the basis of lower surroundings approximations, we define a measure assessing payoffs of the players. For the first player playing for the *Physarum polycephalum* plasmodia, the payoff measure has the form:

$$\Theta^{\bullet} = card(Surr_*(I_t^{\bullet})).$$

For the second player playing for the *Badhamia utricularis* plasmodia, the payoff measure has the form:

$$\Theta^{\circ} = card(Surr_*(I_t^{\circ})).$$

In a more relaxed case, we have respectively:

$$\Theta^{\bullet} = card(Surr_*^{\beta}(I_t^{\bullet})).$$

and

$$\Theta^{\circ} = card(Surr_*^{\beta}(I_t^{\circ})).$$

One can see that the VPRSM approach enables us to set different levels of difficulty of the Go game.

The goal of each player is to maximize its payoff.

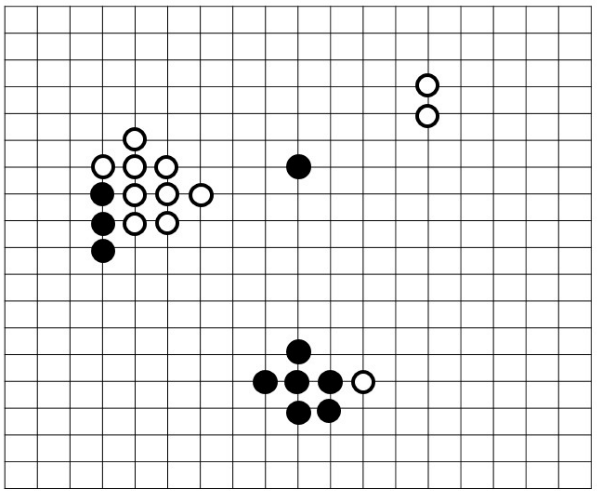


Figure 3: An illustrative configuration of the Go game after several moves

Let us consider an illustrative configuration of the Go game after several moves shown in Figure 3.

In case of a standard definition of rough sets (i.e., the most rigorous case), intersections belonging to lower surroundings approximations $Surr_*(I_t^{\bullet})$ and $Surr_*(I_t^{\circ})$ of $(I_t^{\bullet})$ and (I_t°), respectively, are marked with grey rectangles in Figure 4. It is worth noting that all of the intersections from $Surr_*(I_t^{\bullet})$ and $Surr_*(I_t^{\circ})$ are full generators of the payoffs of the players playing for the *Physarum polycephalum* plasmodia and *Badhamia utricularis* plasmodia, respectively. Hence, we obtain $\Theta^{\circ} = 1$ and $\Theta^{\bullet} = 2$. The second player, playing for the *Badhamia utricularis* plasmodia, wins.

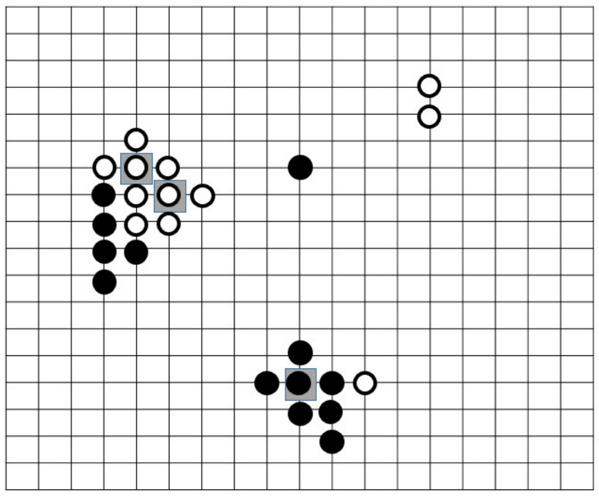


Figure 4: A configuration of the Go game after several moves (payoffs defined on the basis of a standard definition of rough sets)

In case of the VPRSM approach (i.e., a more relaxed case), for $\beta = 0.25$, intersections belonging to lower surroundings approximations $Surr_*^{0.25}(I_t^{\bullet})$ and $Surr_*^{0.25}(I_t^{\circ})$ of $(I_t^{\bullet})$ and (I_t°), respectively, are marked with grey rectangles in Figure 5. It is worth noting that some of intersections from $Surr_*(I_t^\bullet)$ and $Surr_*(I_t^\circ)$ are full quasi-generators of the payoffs of the players playing for the *Physarum polycephalum* plasmodia and *Badhamia utricularis* plasmodia, respectively. Hence, we obtain $\Theta^\circ = 3$ and $\Theta^\bullet = 3$. No player wins.

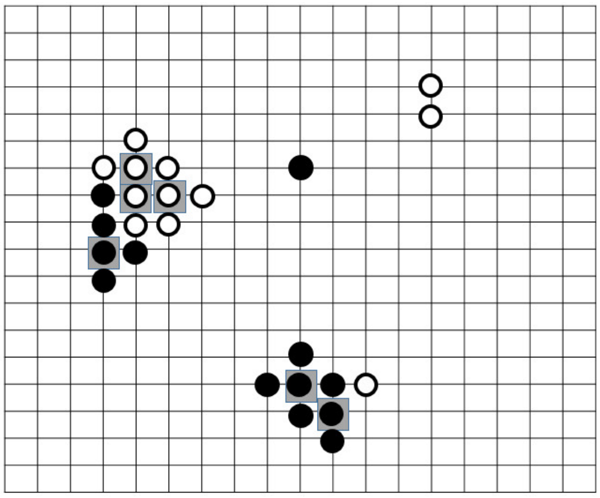


Figure 5: A configuration of the Go game after several moves (payoffs defined on the basis of the VPRSM approach for $\beta = 0.25$)

Let us consider now how we can define strategies in our Go game if we deal with the standard surroundings approximation. A mapping from the intersections belonging to upper surroundings approximations $Surr^*(I_t^\pi)$ at t to the intersections belonging to lower surroundings approximations $Surr_*(I_{t+k}^\pi)$ at $t+k$ is said to be a set of rational strategies of π of radius k to win. A mapping from the intersections belonging to β-lower surroundings approximations $Surr_*^\beta(I_t^\bullet)$ at t to the intersections belonging to the union $Surr_*^\gamma(I_{t+k}^\circ) \cup Surr_*^\beta(I_{t+k}^\bullet)$ at $t+k$, where $0 < \beta \leq \gamma$ and $card(Surr_*^\beta(I_t^\bullet)) \leq card(Surr_*^\gamma(I_{t+k}^\circ) \cup Surr_*^\beta(I_{t+k}^\bullet)) < card(Surr_*(I_{t+k}^\bullet))$, is said to be a set of rational strategies of $\circ$ (the player playing for *Badhamia utricularis*) of radius k not to lose. A mapping from the intersections belonging to β-lower surroundings approximations $Surr_*^\beta(I_t^\circ)$ at t to the intersections belonging to the union $Surr_*^\gamma(I_{t+k}^\bullet) \cup Surr_*^\beta(I_{t+k}^\circ)$ at $t+k$, where $0 < \beta \leq \gamma$ and $card(Surr_*^\beta(I_t^\circ)) \leq card(Surr_*^\gamma(I_{t+k}^\bullet) \cup Surr_*^\beta(I_{t+k}^\circ)) < card(Surr_*(I_{t+k}^\circ))$, is said to be a set of rational strategies of $\bullet$ (the player playing for *Physarum polycephalum*) of radius k not to lose.

The agent is rational if (s)he follows one of the rational strategies to win or not to lose in moves. Also, we can define strategies in the Go game if we deal with the VPRSM surroundings approximation. A mapping from the intersections belonging to γ-lower surroundings approximations $Surr_*^\gamma(I_t^\pi)$ at t to the intersections belonging to β-lower surroundings approximations $Surr_*^\beta(I_{t+k}^\pi)$ at $t+k$ is said to be a set of rational β-strategies of π of radius k to win if $\gamma < beta$. A mapping from the intersections belonging to γ-lower surroundings approximations $Surr_*^\gamma(I_t^\bullet)$ at t to the intersections belonging to the union $Surr_*^\delta(I_{t+k}^\circ) \cup Surr_*^\gamma(I_{t+k}^\bullet)$ at $t+k$, where $0 < \beta \leq \gamma$ and $0 < \beta \leq \delta$ and $card(Surr_*^\gamma(I_t^\bullet)) \leq card(Surr_*^\delta(I_{t+k}^\circ) \cup Surr_*^\gamma(I_{t+k}^\bullet)) < card(Surr_*^\beta(I_{t+k}^\bullet))$, is said to be a set of rational β-strategies of $\circ$ (the player playing for *Badhamia utricularis*) of radius k not to lose. A mapping from the intersections belonging to γ-lower surroundings approximations $Surr_*^\gamma(I_t^\circ)$ at t to the intersections belonging to the union $Surr_*^\delta(I_{t+k}^\bullet) \cup Surr_*^\gamma(I_{t+k}^\circ)$ at $t+k$, where $0 < \beta \leq \gamma$ and $0 < \beta \leq \delta$ and $card(Surr_*^\gamma(I_t^\circ)) \leq card(Surr_*^\delta(I_{t+k}^\bullet) \cup Surr_*^\gamma(I_{t+k}^\circ)) < card(Surr_*^\beta(I_{t+k}^\circ))$, is said to be a set of rational β-strategies of $\bullet$ (the player playing for *Physarum polycephalum*) of radius k not to lose.

The agent is β-rational if (s)he follows one of the rational β-strategies to win or not to lose in moves.

4. A SOFTWARE TOOL

In [13], we described selected functionality of the current version of a new software tool, called *PhysarumSoft*, developed by us for programming *Physarum* machines and simulating *Physarum* games. In this section, we recall important features of this tool, as well as present additional information about the part of the tool responsible for simulating a rough set version of the Go game.

PhysarumSoft was designed for the Java platform. A general structure of this tool is shown in Figure 6. We can

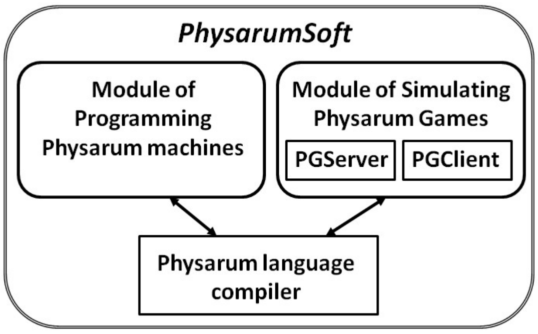

PhysarumSoft
Module of Programming Physarum machines
Module of Simulating Physarum Games
PGServer
PGClient
Physarum language compiler

Figure 6: A general structure of *PhysarumSoft*

distinguish three main parts of *PhysarumSoft*:

- *Physarum* language compiler.
- Module of programming *Physarum* machines.
- Module of simulating *Physarum* games.

The main features of *PhysarumSoft* are the following:

- Portability. Thanks to the Java technology, the created tool can be run on various software and hardware platforms. In the future, the tool will be adapted for platforms available in mobile devices and as a service in the cloud.
- User-friendly interface.
- Modularity. The project of *PhysarumSoft* and its implementation covers modularity. It makes the tool extend easily in the future.

To simulate games on *Physarum* machines, we are developing a special module of *PhysarumSoft* called the *Physarum* game simulator. This module works under the client-server paradigm. A general structure of the *Physarum* game simulator is shown in Figure 7.

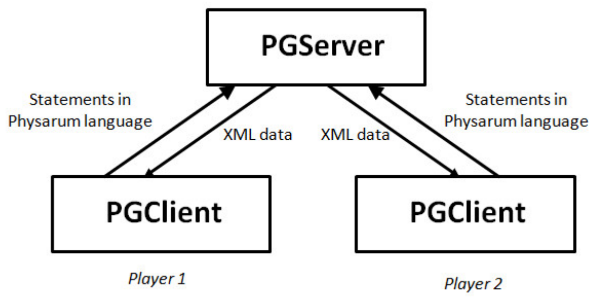

PGServer
Statements in Physarum language
XML data
XML data
Statements in Physarum language
PGClient
PGClient
Player 1
Player 2

Figure 7: A general structure of the *Physarum* game simulator

In *PhysarumSoft*, communication between clients and the server is realized through text messages containing statements of the *Physarum* language. The *Physarum* language, is a new object-oriented programming language, utilizing the prototype-based approach (cf. [3]), designed by us to program *Physarum* machines, i.e., to set the spatial distribution (configuration) of stimuli (attractants, repellents) controlling propagation of protoplasmic veins of the plasmodium. For a detailed description of the *Physarum* language, we refer the reader to [6], [9], and [15].

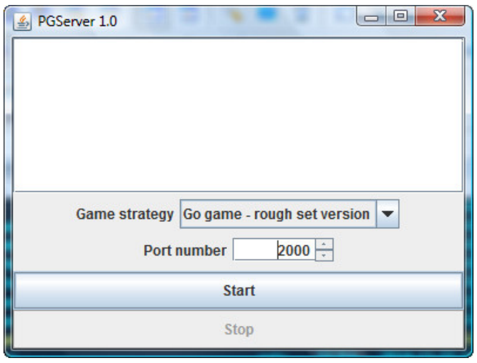
The server sends to clients information about the current configuration of the *Physarum* machine (localization of the original points of *Physarum polycephalum* and *Badhamia utricularis*, localization of stimuli, as well as a list of edges, corresponding to veins of plasmodia, between active points) through the XML file. Each original point of the plasmodia of *Physarum polycephalum* or the plasmodia of *Badhamia utricularis* and each attractant occupied by these plasmodia is called an active point in the *Physarum* machines.

The server-side application of the *Physarum* game simulator is called *PGServer*. The main window of *PGServer* is shown in Figure 8. In this window, the user can:

- select the port number on which the server listens for connections,
- start and stop the server,
- set the game:
  - a *Physarum* game with strategy based on stimulus placement (see [7]),
  - a *Physarum* game with strategy based on stimulus activation (see [7]),
  - a rough set version of the Go game (see Section 3),
- shadow information about actions undertaken.

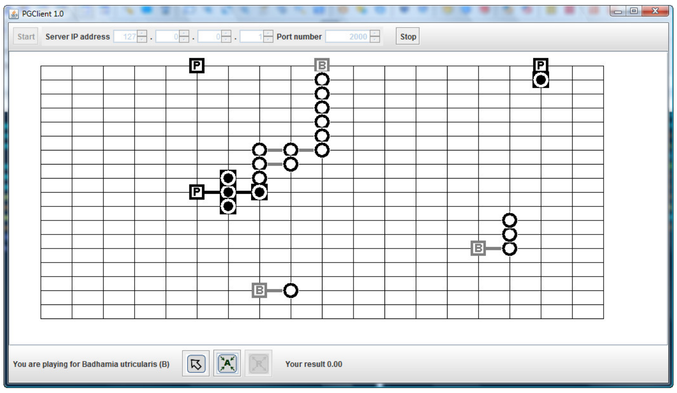
The client-side application of the *Physarum* game simulator is called *PGClient*. The main window of *PGClient* is shown in Figure 9. In case of the Go game, the user can:

- set the server IP address and its port number,
- start the participation in the game,
- put attractants at the vacant intersections of a board,
- monitor the current state of the game as well as the current assessment of payoffs.



Game strategy: Go game - rough set version
Port number: 2000

Figure 8: The main window of *PGServer*



Server IP address 127 . 0 . 0 . 1 Port number 2000

You are playing for Badhamia utricularis (B)

Your result 0.00

Figure 9: The main window of *PGClient*

5. CONCLUSIONS

A new version of the Go game, based on rough set theory and implemented on the *Physarum* machines, has been presented. The presented version is an antagonistic game. The locations of black stones are understood as intersections occupied by plasmodia of *Physarum polycephalum* and the locations of white stones are understood as intersections occupied by plasmodia of *Badhamia utricularis*. The main aim of our further research is to propose a coalition game based on a rough set assessment of payoffs. Moreover, we will extend the spectrum of measures by applying various rough set approaches.

6. ACKNOWLEDGMENTS

This research is being fulfilled by the support of FP7-ICT-2011-8.

7. REFERENCES

[1] A. Adamatzky. *Physarum Machines: Computers from Slime Mould*. World Scientific, 2010.

[2] A. Adamatzky, V. Erokhin, M. Grube, T. Schubert, and A. Schumann. Physarum Chip Project: Growing computers from slime mould. *International Journal of Unconventional Computing*, 8(4):319–323, 2012.

[3] I. Craig. *Object-Oriented Programming Languages: Interpretation*. Springer-Verlag, London, 2007.

[4] D. Dubois and H. Prade. Rough fuzzy sets and fuzzy rough sets. *International Journal of General Systems*, 17(2-3):191–209, 1990.

[5] J. Kim and S. Jeong. *Learn to Play Go*. Good Move Press, New York, 1994.

[6] K. Pancerz and A. Schumann. Principles of an object-oriented programming language for Physarum polycephalum computing. In *Proceedings of the 10th International Conference on Digital Technologies (DT'2014)*, pages 273–280, Zilina, Slovak Republic, 2014.

[7] K. Pancerz and A. Schumann. Rough set description of strategy games on Physarum machines. In A. Adamatzky, editor, *Advances in Unconventional Computing*. Springer International Publishing, 2016. to appear.

[8] K. Pancerz and A. Schumann. Rough set models of Physarum machines. *International Journal of General Systems*, 44(3):314–325, 2015.

[9] K. Pancerz and A. Schumann. Some issues on an object-oriented programming language for Physarum machines. In R. Bris, J. Majernik, K. Pancerz, and E. Zaitseva, editors, *Applications of Computational Intelligence in Biomedical Technology*, volume 606 of *Studies in Computational Intelligence*, pages 185–199. Springer International Publishing, 2016.

[10] Z. Pawlak. *Rough Sets. Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, 1991.

[11] Z. Pawlak and A. Skowron. Rudiments of rough sets. *Information Sciences*, 177:3–27, 2007.

[12] A. Schumann. Syllogistic versions of Go games on Physarum polycephalum. In A. Adamatzky, editor, *Advances in Physarum Machines*. Springer International Publishing, 2016. to appear.

[13] A. Schumann and Pancerz. PhysarumSoft - a software tool for programming Physarum machines and simulating Physarum games. In M. Ganzha, L. Maciaszek, and M. Paprzycki, editors, *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems (FedCSIS'2015)*, pages 607–614, Lodz, Poland, 2015.

[14] A. Schumann and Pancerz. Roughness in timed transition systems modeling propagation of plasmodium. In D. Ciucci, G. Wang, S. Mitra, and W.-Z. Wu, editors, *Rough Sets and Knowledge Technology*, volume 9436 of *Lecture Notes in Artificial Intelligence*. Springer International Publishing, 2015.

[15] A. Schumann and K. Pancerz. Towards an object-oriented programming language for Physarum polycephalum computing. In M. Szczuka, L. Czaja, and M. Kacprzak, editors, *Proceedings of the Workshop on Concurrency, Specification and Programming (CS&P'2013)*, pages 389–397, Warsaw, Poland, 2013.

[16] A. Schumann and K. Pancerz. Interfaces in a game-theoretic setting for controlling the plasmodium motions. In *Proceedings of the 8th International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS'2015)*, pages 338–343, Lisbon, Portugal, 2015.

[17] A. Schumann, K. Pancerz, A. Adamatzky, and M. Grube. Bio-inspired game theory: The case of Physarum polycephalum. In *Proceedings of the 8th International Conference on Bio-inspired Information and Communications Technologies (BICT'2014)*, Boston, Massachusetts, USA, 2014.

[18] W. Ziarko. Variable precision rough set model. *Journal of Computer and System Sciences*, 46(1):39–59, 1993.