2A2-2 Strategic Behavior Generation with Cognitive Distance in Two-Player Games

Kosuke Sekiyama Ricardo Carnieri Toshio Fukuda

Dept. of Micro-Nano Systems Engineering, Nagoya University, Japan
Email: {sekiyama,carnieri,fukuda}@mein.nagoya-u.ac.jp

Abstract— In game theoretical approaches to multi-agent systems, a payoff matrix is often given a priori and used by agents in action selection. By contrast, in this paper we approach the problem of decision making by use of the concept of cognitive distance, which is a notion of the difficulty of an action perceived subjectively by the agent. As opposed to ordinary physical distance, cognitive distance depends on the situation and skills of the agent, ultimately representing the perceived difficulty in performing an action given the current state. We show how an agent can learn its cognitive distance parameters by estimating and observing the outcomes of its actions. This learning algorithm is then applied to two-player game scenarios.

Key Words: multi-agent, cognitive distance, game theory, markov games

1 Introduction

Environmental uncertainty in multi-agent domains can be considered as being of three different forms: a) the transitions in the domain itself might be non-deterministic; b) agents might not know the actions of other agents; c) agents might not know the outcomes of their own actions [1].

In this work, we investigate the case when the agents have limited physical capabilities, which causes limited precision in their actions. We consider the scenario of two agents playing a simplified tennis game where the actions have uncertain outcomes. These uncertainties originate from the limited physical capabilities of the agents. Because there is more than one agent and more than one state, this game is neither a Markov Decision Process (MDP), nor a matrix game; therefore we use the framework of *Markov games* (also called *stochastic games*), as described in [2, 3].

The perceived distance has an important function in decision making and action selection. It is assumed that every action has a target, which may or may not have a physical meaning depending on the nature of the domain. In sports, in particular, the target of an action often is a physical location, such as in passing the ball to a teammate in soccer, basketball, and countless other sports, or as in hitting the ball to the service box in a tennis serve. In basketball, although the difficulty of a shot clearly increases with basket-player distance, this is not the only factor that influences the difficulty. For example, different players have different skills and will perform differently even at identical shot distance; and shooting during training is very different from shooting during an actual match, when the players are under considerable pressure.

The total effect of these factors on an action's outcome can be thought of as the perceived action difficulty. We call the perceived action difficulty *cognitive distance* [4]. Figure 1 illustrates the difference between physical distance and cognitive distance in a scenario where the agents are directed. Suppose that the task is to pass a ball to the other agent. In most sports where this situation might occur, the passing skill of the player depends on the relative angle of the direction the player is facing and the pass direction. In soccer, for example, the majority of passes are done in the direction the player is facing, or at a small angle from it. Because players are bound by physical constraints [4], it is much harder to pass the ball backward than forward. Since the agent cannot directly sense its own skills and physical constraints, it must use the difference between predicted average action outcome and actual action outcomes to assess its cognitive distance [4].

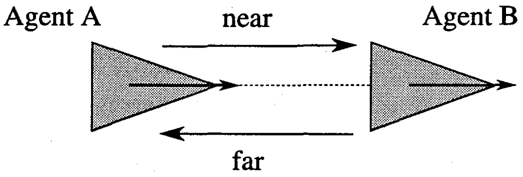


Fig.1 Cognitive distance, as opposed to physical distance, depends on the situation

We focus on two questions related to uncertainty in action outcomes: how to learn the action-outcome relationship, and how to perform action selection given this relationship. Our objective is to construct a framework of action selection in two-player games using cognitive distance to express the action-outcome map.

This paper is organized as follows. In Section 2, we describe the tennis game model used and define cognitive distance. Section 3 describes how strategic behavior is generated in this game using the cognitive distance. In Section 4, we describe how an unknown skill parameter can be learned by the player and show how wrong beliefs of the value of this parameter influence performance. In Section 5, we present simulation results using the tennis game model. Finally, Section 6 contains concluding remarks on this work.

2 Model

A model showing that professional tennis players perform similar to the mixed strategy equilibrium is given in [5]. In that work, it is assumed that every point in a tennis match is played as a 2 × 2 normal-form game, by focusing on the actions chosen by server and receiver. The play after the serve is not modeled, and instead a reduced form representation of it is used to give the expected payoffs.

By contrast, in this work we attempt to model the action selection process during the entire play, from serve until a player scores. We consider two agents playing a tennis-like two-dimensional game (figure 4). By two-dimensional we mean that the ball altitude is ignored for receiving purposes. Players serve alternately to any region on the opponent side. The score of a player is incremented by one when it wins the current point game. The score is simply incremental, meaning that there are no games and no sets. The players have three behaviors: target selection and hitting, moving after hitting and intercepting the ball to receive (figure 2).

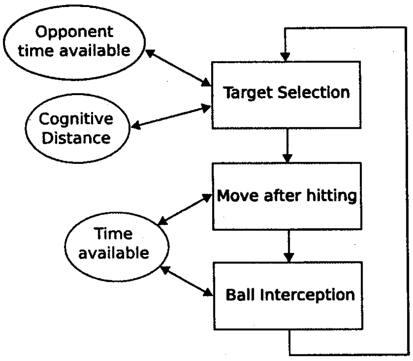


Fig.2 Block diagram of the behavior generation algorithm using cognitive distance in the tennis game

We assume that four factors influence the probability that a hit action is successful: agent skill, hit target, hit strength, and the time available for preparing to hit (see figure 6 for a list of the most important symbols and their meanings). Player skill is assumed invariable during play. Hit target and hit strength are chosen by the hitting player and constitute the *hitting action.* The last factor, time available to hit, is defined as the time during which the player was still before hitting. It is used in action selection to encode both the fact that the agent tries to hit the ball to places out of reach of its opponent, and the fact that the agent positions itself in order to have as much time to prepare to hit as possible.

We assume that the players have limited precision when hitting the ball. This is denoted by a bivariate normal distribution of the landing point of the ball given hit target $P_T(t) = (\mu_x, \mu_y)$:

$$pdf_T(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\left(\frac{(x-\mu_x)^2}{2\sigma_x^2}+\frac{(y-\mu_y)^2}{2\sigma_y^2}\right)\right) \tag{1}$$

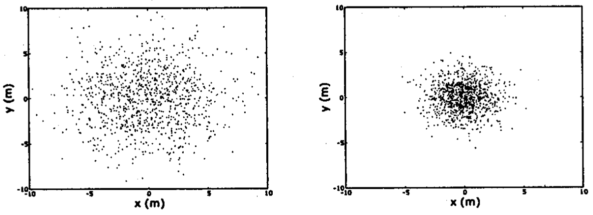
where $\sigma_x = \sigma_y = \sigma$ is the standard deviation[1]. The higher the σ, the less the probability that the ball will go where intended. In other words, σ represents the lack of precision of the player. We assume it depends on the time available t_a to prepare for hitting and on the hitting speed v_h. Intuitively, the longer the time available to hit t_a and the slower the hitting speed v_h, the smaller the deviation, and therefore the more precise the hit outcome is.

The hitting precision also depends on parameters representing the skill of the player, as shown in eq.(2). The hitting skill parameter σ_0 is the minimum deviation achievable by the player in optimal conditions; the smaller the σ_0, the more precise the player is overall. The parameters t_{ac} and t_{as} in eq.(3) control the shape of the sigmoid function $\mathcal{T}(t_a)$, which dictates how the hitting precision changes with t_a; t_{ac} represents how much t_a is necessary for the hit precision to be average, and t_{as} represents how abruptly the hitting precision changes with t_a. Likewise, v_{hc} and v_{hs} in eq.(4) control the shape of the sigmoid function $\mathcal{V}(v_h)$, with v_{hc} representing the hitting speed at which precision is average, and v_{hs} representing how abruptly the hitting precision changes with v_h. Figure 3 shows scatter plots of hit outcomes for different t_a.

$$\sigma = \frac{\sigma_0}{\mathcal{T}(t_a)(1-\mathcal{V}(v_h))} \tag{2}$$

$$\mathcal{T}(t_a) = \frac{1}{1+\exp(-\frac{t_a-t_{ac}}{t_{as}})} \tag{3}$$

$$\mathcal{V}(v_h) = \frac{1}{1+\exp(-\frac{v_h-v_{hc}}{v_{hs}})} \tag{4}$$



(a) $t_a = 0.4$

(b) $t_a = 0.8$

Fig.3 Scatter plot of 1000 hits with target $(x, y) = (0, 0)$, global hitting skil $\sigma_0 = 0.4$, and hitting speed $v_h = 12$

Let $P_H(t)$, $P_T(t)$ and $P_R(t)$ denote the positions of hitting player, hit target and receiving player, respectively, at time t. A hitting action (P_H, P_T, v_h) is characterized by hit origin P_H, hit target P_T and hit speed v_h. The actual outcome of the hit is drawn from the probability distribution function described in eq.(1). Note that, by the definitions above, the hit outcome does not depend on the hit origin P_H, which is assumed for the sake of simplicity.

A hit action is successful if the ball bounces inside the opposite side of the court (figure 4). Success, here, does

[1] We assume that the random variables x and y are uncorrelated and have the same standard deviation.

not mean scoring a point, or even sending the ball to a desired target, but rather hitting the ball to any valid position on the side of the opponent. In other words, *success* is the opposite of *fault.* Let $g_T(t_a, v_h)$ denote the success probability of hitting the ball to target P_T with speed v_h, given that the player had time available t_a to prepare the hit. Likewise, let $f_T(t_a, v_h) = 1 - g_T(t_a, v_h)$ denote the fault probability. The success probability g_T can be calculated by integrating $pdf_T(x, y)$ over the region S of the half-court of the opponent as in eq.(5). This is done by numerically approximating the cumulative bivariate distribution.

$$g_T(t_a, v_h) = \int_{\mathbf{x} \in S} pdf_T \ d\mathbf{x} \tag{5}$$

We define the *cognitive distance* D_T of the hit action as

$$D_T(t_a, v_h) \equiv f_T(t_a, v_h) = 1 - g_T(t_a, v_h) \tag{6}$$

i.e., the cognitive distance of a hit action equals its fault probability. This agrees with the concept that cognitive distance expresses the difficulty in accomplishing an action.

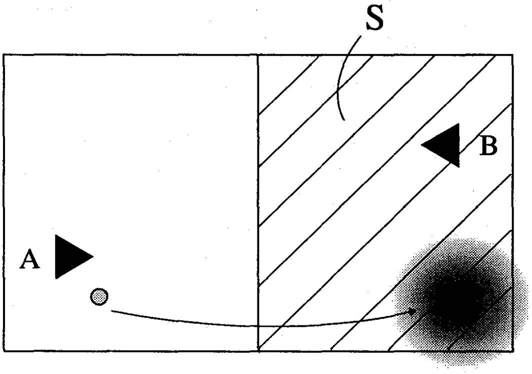


Fig.4 The fault probability is calculated by integrating the cumulative bivariate distribution over the region S. Darker areas are more likely outcomes.

The cognitive distance, or fault probability, of a hit action can be considered the *immediate cost* associated with performing the action. It is an immediate cost because it is the probability that the player might commit a fault. However, it is different from the *expected cost,* which we define as the probability of losing the point game, because even if the hit is successful, the player might still lose the point game. In order to estimate the expected cost associated with a hit action, the player must calculate the receive fault probability of the opponent given that his own hit was successful. This would require knowing the opponent skill parameters σ_0, t_{ac}, t_{as}, v_{hc} and v_{hs} (see figure 6 for the list of symbols), as well as how much time the opponent will have to prepare for receiving the ball.

The *time available* is defined as the slack time the player has to prepare for receiving the ball. Intuitively, the larger the slack, the more accurate and powerful the reception can be. Therefore, the hitting player should choose a target such as to minimize both the immediate cost *and* the time available to the opponent for reception.

3 Strategic Behavior Generation

3.1 Ball interception

In this model the only randomness lies in the actual hit outcomes. Once the ball is hit, its trajectory is exactly determined and has a fixed speed until it is hit again by the other player. Therefore, if the receiving player can observe the incoming ball direction at all times and with no noise, it can determine the optimal position P_R^* from which to receive.

The optimal receiving position P_R^* can be calculated by assuming that the receiver approaches the ball with a constant bearing angle [6] (Figure 5). The time-optimal trajectory for interception is given by

$$\phi = \arcsin\left(\frac{v_B \ \sin\beta}{v_R}\right) \tag{7}$$

where v_R and v_B are the receiver and ball velocity, respectively. If the receiver is too slow ($v_R < v_B \ \sin\beta$) there is no solution to eq.(7) and the ball cannot be intercepted. If $v_R > v_B \ \sin\beta$, then there are two solutions to eq. 7 but only one of these makes the receiver approach the ball.

The angle ϕ_{min} that requires the minimum speed and still allows interception is found by letting $v_R = v_{min} = v_B \ \sin\beta$. The angle ϕ_{min} requires the least speed from the receiver compared to any other angle. Therefore, if the receiving player moves according to ϕ_{min} with its maximum speed and stops when it gets to P_R^*, it will have maximized the time available t_a to prepare for the hit.

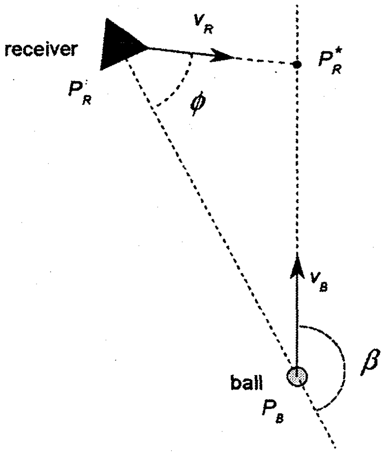


Fig.5 Time-optimal target interception

3.2 Hit target selection

The hitting player must choose a target that simultaneously has a small fault probability f_T and imposes a small t_a on the opponent. A player that attempts to minimize the fault probability when choosing a target will always hit to the center (which, incidentally, is what many unexperienced human players tend to do). On the other hand, a player that attempts to minimize the imposed ta on the opponent will always try to hit to the corner furthest away from the opponent. Neither of these two extremes are good playing strategies: the former tends to miss chances to settle the point, and the latter tends to miss too many hits. Clearly, a balance must exist.

Let l_T denote the cost associated with hitting the ball to target P_T. We define

$$l_T(t_a, f_T) = \gamma\tau + (1-\gamma)f_T \tag{8}$$

$$\tau = \frac{t_a - min_{ta}}{max_{ta} - min_{ta}} \tag{9}$$

where τ is the normalized time available t_a for reception imposed on the opponent and f_T is the fault probability when hitting to target P_T, as defined in eq.(6). Here, γ represents how much the agent is willing to risk. The two extremes mentioned earlier are $\gamma = 0$ (minimum risk, always hit to the center), and $\gamma = 1$ (maximum risk, always hit to the corners). A balanced strategy is obtained by using an intermediate value for γ.

The hit target P_T^* selected is the one that minimizes the cost:

$$P_T^* = \underset{P_T}{\operatorname{argmin}}\, l_T(t_a, f_T) \tag{10}$$

3.3 Moving after hitting

In the interval between hitting the ball and the opponent receiving it, where should the player move to? In real tennis it is common to see professional players running to the center of the baseline after hitting. This is intuitive to understand as they do not know where the opponent will hit next. However, the players also go up to net, and do not always stay near the center. This is not an easy matter, for the best position depends not only on the skills of both players, but also on their strategies.

In this work, we adopt a solution that is suboptimal but still allows for realistic game play. Let A denote the player who just hit and B the player who is receiving. If B chooses the reception point P_R^* such as to maximize its t_a, as described in Section 3.1, then A knows the ball will be received from P_R^* the moment it observes the outcome of its hit. Let $P_D(t)$ denote the destiny chosen by A. It can search for the best P_D by calculating the t_a imposed on itself, considering that B will hit from P_R^* and try to minimize the t_a imposed on A. All A has to do, then, is choose P_D^* such that the minimum t_a imposed on itself is maximized; this is known as the *maximin* strategy [7].

4 Cognitive distance learning

In order to calculate the cognitive distance D_T used in hit target selection, the player must know its skill parameters σ_0, t_{ac}, t_{as}, v_{hc} and v_{hs}. An algorithm for learning a cognitive distance parameter using Q-learning [8] was described in [4]. It consists in assessing Q-values for different values of the parameter by observing actual outcomes of the action. First, it selects a target and a value of the cognitive distance parameter and derives the error between estimated success probability and the actual average success probability over N outcomes of the action. A reward value derived from the error is then used to update the Q-value of the currently used value of the parameter. The target is then updated and the process repeated until the Q-values converge.

Symbol	Meaning	Type
t_a	time available to prepare hit	variable
v_h	hitting speed	variable
σ_0	player hitting skill	parameter
t_{ac}	characteristic t_a	parameter
t_{as}	slope of $\mathcal{T}(t_a)$	parameter
v_{hc}	characteristic v_h	parameter
v_{hs}	slope of $\mathcal{V}(v_h)$	parameter
$P_H(t)$	hit position	variable
$P_T(t)$	hit target position	variable
$P_T^*(t)$	optimal target position	variable
$P_R(t)$	receiver position	variable
$P_R^*(t)$	optimal receiving position	variable
$P_D(t)$	player destiny after hitting	variable
$P_D^*(t)$	optimal destiny after hittting	variable
$P_B(t)$	ball position	variable
$v_R(t)$	receiver speed	variable
$v_B(t)$	ball speed	variable
g_T	hit success probability	variable
f_T	hit fault probability	variable
D_T	cognitive distance	variable
ϕ	interception angle	variable
β	angle between player–ball line and ball trajectory	variable
l_T	hit cost	variable
γ_1	risk strategy of player 1	parameter
γ_2	risk strategy of player 2	parameter

Fig.6 Symbol list

We used the algorithm described in [4] with a minor simplification; namely, we used random instead of directed target selection. Although there is more than one skill parameter, we only considered the scenario where t_{ac} is unknown and the other parameters are known. Figure 7 describes the Q-learning algorithm used. A typical run of the algorithm is shown in figure 8.

The importance of having a correct belief of the parameter t_{ac} can be seen in figure 9. Both players had exactly the same skill parameters and used the same strategy ($\gamma_1 = \gamma_2 = 0.3$), but whereas player 1 used the actual value of t_{ac} when estimating fault probability, player 2 used a belief of the value of t_{ac}. For small values of t_{ac} belief, the winning ratio of player 1 is higher, because player 2 is underestimating its fault probability and therefore performing risky hits, i.e., targeting the borders even when the fault probability is high. For belief $t_{ac} = 0.8$, player 2 is only slightly underestimating its fault probability and performs as well as player 1. For larger values of t_{ac} belief, including the correct value $t_{ac} = 1.0$, the winning ratio is also close to 50%. Overestimating the fault probability in this situation does not incur in reduced performance for player 2, since both players are using a risk-avoiding strategy.

```
begin
  initialize $Q(t_{ac}) = 0 \quad \forall\, t_{ac}$
  repeat
    select target $P_T$
    select $t_a$
    select $v_h$
    select $t_{ac}$ with exploration
    calculate $\hat{g_T}(t_a, v_h)$
    repeat N times
      execute hit with target $P_T$
      if hit successful:
        $u_n = 1$
      else:
        $u_n = 0$
    $G = \frac{1}{N}\sum_{n=1}^{N} u_n$
    $E = G - \hat{g_T}$
    $Q(t_{ac}) \leftarrow (1-\alpha)Q(t_{ac}) + \alpha r(E)$
end
```

Fig.7 Q-learning algorithm for learning skill parameter t_{ac}

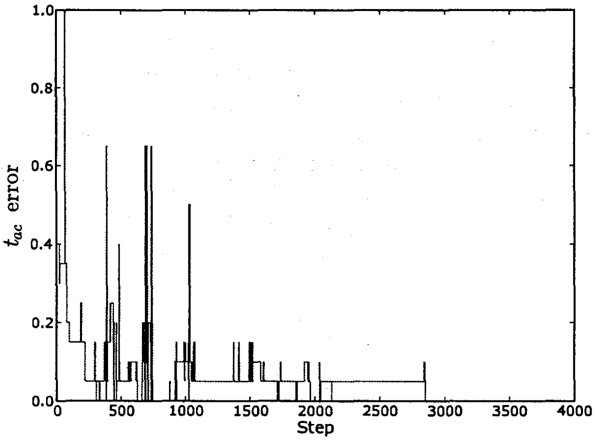


Fig.8 Error of learned parameter t_{ac}. Each time step consists in randomly selecting a target and performing $N = 25$ hits. Simulation conditions are $\alpha = 0.1$, $t_{ac} = 1.0$.

5 Simulation Results

We conducted a series of simulations to verify the validity of the action selection algorithms and the relevance of cognitive distance in strategic behavior generation. All results were obtained by running the simulation until 1000 points were scored. Players served alternately in order to average out serve advantages (or disadvantages). Hitting speed was fixed at $v_h = 12$ and both players had the same hitting skill σ_0. Figure 10 shows a snapshot of the simulation.

In the first simulation (figure 11), player 1 used a fixed strategy of $\gamma_1 = 0.3$ while player 2 used the entire range of strategies, from risk-avoiding ($\gamma_2 = 0.0$) to risk-seeking ($\gamma_2 = 1.0$).

For skilled players ($\sigma_0 = 0.2$), the winning ratios were close to 50%; when $\gamma_2 > \gamma_1$, player 1 scored somewhat more frequently.

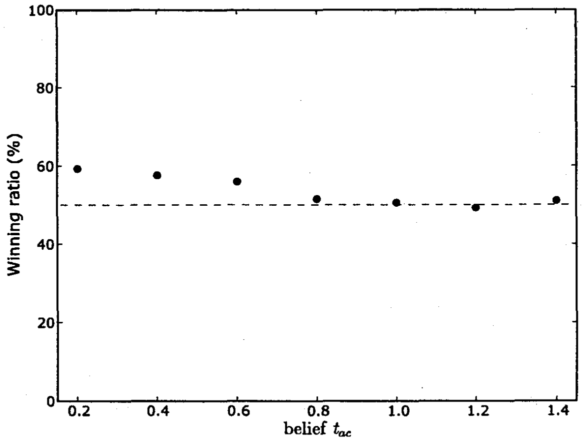


Fig.9 Winning ratio of player 1 for different player 2 beliefs of t_{ac}. Simulation conditions are $t_{ac} = 1.0$, $\gamma_1 = \gamma_2 = 0.3$, $\sigma_0 = 0.2$.

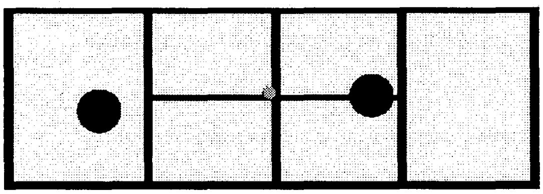


Fig.10 Snapshot of the simulation

For average players ($\sigma_0 = 0.4$), player 1 clearly beat player 2 when $\gamma_2 > \gamma_1$, meaning that more aggressive strategies lead to too many faults when the uncertainty in hit outcomes is moderate.

For unskilled players ($\sigma_0 = 0.8$), the situation above repeats, except that the winning margin is not so wide. This reduced winning ratio for player 1 compared to the $\sigma_0 = 0.4$ case can be explained by noting that, since both players are unskilled, they are committing many faults already, and a more risky strategy has a reduced effect on the winning ratio.

In the second simulation (figure 12), the skill of the players was fixed at $\sigma_0 = 0.2$, and the winning ratio for player 1 was obtained for different pairs of strategies.

A very conservative strategy ($\gamma_1 = 0.0$) lost to slightly more aggressive strategies ($\gamma_2 = 0.1, \gamma_2 = 0.2$), but had an edge over very aggressive ones ($\gamma_2 > 0.5$).

When slightly aggressive ($\gamma_1 = 0.3$), player 1 obtained a winning ratio of at least 50%, usually higher, for any strategy employed by the opponent. This indicates that $\gamma = 0.3$ is a dominating strategy.

When using a very risky strategy ($\gamma_1 = 1.0$), player 1 is clearly at a disadvantage, especially when the opponent is only slightly aggressive ($\gamma_2 = 0.1, \gamma_2 = 0.2$). When player 2 also employs a risky strategy, however, the winning ratio is close to 50%.

These results show that there is a balance between safe strategies and risky strategies. The largest expected payoff can only be obtained by correctly assessing the fault probability, which requires knowing the skill pa-

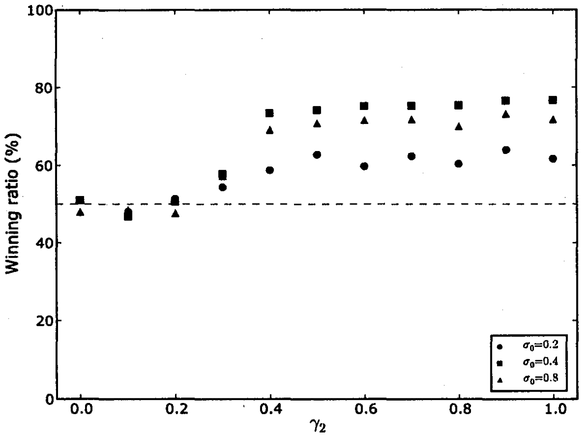


Fig.11 Winning ratio of player 1 in function of γ_2 for different values of σ_0,γ_1 = 0.3.

rameters as discussed in Section 4. Even in matches between players with exactly the same skills, different strategies might yield very different payoffs.

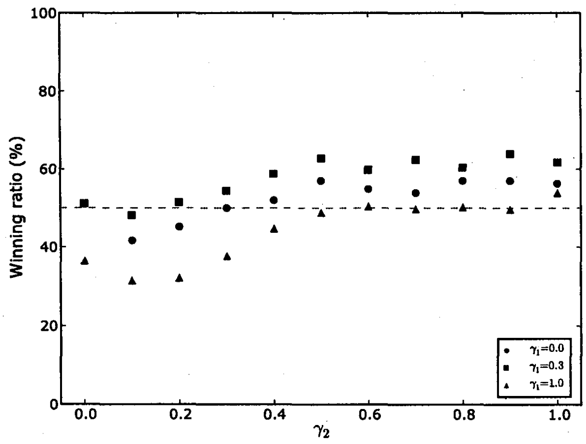


Fig.12 Winning ratio of player 1 in function of γ_2 for different values of γ_1 , σ_0 = 0.2.

6 Conclusion

We have discussed how the problem of action selection in a two-player game with uncertain action outcomes can be formalized in a framework using cognitive distance. Uncertainty in the actions of embedded agents arises because the agents are limited by physical constraints. In this context, cognitive distance is a measure of the difficulty in accomplishing an action given the current situation. By using the cognitive distance an agent can assess, given the current situation, how likely its actions are to succeed, and perform action selection based on this success probability.

The approach has been tested in simulation of a tennis-like game. We described how cognitive distance can be used for hit target selection and that a trade off is needed between trying to score a point and trying not to commit a fault.

The next steps on our research agenda include extending the learning algorithm to multiple parameters. While we have only done experiments of skill parameter learning with one unknown parameter, there it nothing in principle that precludes the use of the same algorithm for learning multiple unknown parameters. Another aspect that was not approached in this work was learning the skill parameter *during* play. The algorithm described requires selecting a target and averaging the outcomes of N hits in succession, which is not possible during play. The reason is that, in its present form, the Q-learning algorithm only stores the Q-values and the outcomes of the last batch of hit trials. Learning during play, however, would require storing past targets and their respective conditions, t_a and v_h.

References

[1] Stone, P. and Veloso, M. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, July 2000.

[2] Littman, M. Markov games as a framework for multi-agent reinforcement learning. In W. W. Cohen and H. Hirsh, editors, *Proceedings of the Eleventh International Conference on Machine Learning (ML-94)*, pages 157–163, New Brunswick, NJ, 1994. Morgan Kauffman Publishers, Inc.

[3] Bowling, M. and Veloso, M. An analysis of stochastic game theory for multiagent reinforcement learning. Technical report CMU-CS-00-165, Computer Science Department, Carnegie Mellon University, 2000.

[4] Sekiyama, K. and Yamamoto, T. Learning and adaptation of cognitive distance for directed multi-agent systems. SCIS & ISIS 2006. To appear.

[5] Walker, M. and Wooders, J. Minimax play at wimbledon. *The American Economic Review*, 91(5):1521–1538, December 2001.

[6] Ghose, K., Horiuchi, T. K., Krishnaprasad, P. S. and Moss, C. F. Echolocating bats use a nearly time-optimal strategy to intercept prey. *PLoS Biol*, 4(5):e108, May 2006.

[7] Walker, M. and Wooders, J. Equilibrium play in matches: Binary markov games. mimeo, University of Arizona, July 2000.

[8] Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.