Learning of Winning Strategies for Terminal Games with Linear-Size Memory

Thomas Böhme *
Institut für Mathematik
Technische Universität Ilmenau
Ilmenau, Germany

Frank Göring †
Fakultät für Mathematik
Technische Universität Chemnitz
Chemnitz, Germany

Zsolt Tuza ‡
Computer and Automation Institute
Hungarian Academy of Sciences
Budapest, Hungary

Herwig Unger §
Fachbereich Informatik
Universität Rostock
Rostock, Germany

Latest update on 19–09–2006

Abstract

We prove that if one or more players in a locally finite positional game have winning strategies, then they can find it for themselves, not losing more than a bounded number of plays and not using more than a *linear-size memory*, independently of the strategies applied by the other players. We design two algorithms for learning how to win. One of them can also be modified to determine a strategy that achieves a draw, provided that no winning strategy exists for the player in question but with properly chosen moves a draw can be ensured from the starting position. If the drawing- or winning strategy exists, then it is learnt after no more than a *linear number of plays* lost.

*E-mail address: tboehme@theoinf.tu-ilmenau.de

†E-mail address: frank.goering@mathematik.tu-chemnitz.de

‡Also affiliated with the Department of Computer Science, University of Veszprém, Hungary. Research supported in part by the grant OTKA T-049613. E-mail address: tuza@sztaki.hu

§E-mail address: hunger@informatik.uni-rostock.de

1 Introduction

For determining a winning strategy in a positional game, several advanced methods have been developed. Some prominent examples are the Grundy function (e.g., [3, 8, 9]) and its generalizations (e.g., [5]) as well as the decomposition of a game into a sum of smaller games (e.g., [13]). An almost complete overview on the existing literature about combinatorial game theory can be found in the dynamic survey [6].

Those techniques, however, assume complete information on the structure of the game in question.

In this paper we are concerned with the situation where a player has a winning strategy from the starting position but, lacking a global overview on the entire game, he/she only knows the (local) alternatives of his/her possible moves in each position of the game. In the main results of Section 4 we prove that a winning strategy can be learnt after a finite number of plays lost, even when only this rather limited information is accessible, no matter how the other players play the game. It does not mean that in an infinite sequence of plays the winning strategy surely is found; but nevertheless it does mean that the player *eventually wins all subsequent plays*.

On the road to permanent winning, only the number of plays lost is bounded, while the number of intermediate plays won may depend on the strategies of the other players and it has no universal upper bound. We prove, however, that if in every play the strategy of each player is uniquely determined by their strategies played and scores achieved during the preceding k plays (where k is an arbitrary but fixed positive integer), then the player with a winning strategy can find a way within a *bounded number of plays* (counting both the ones won and lost), how to win all plays afterwards.

Methods of finding a winning strategy do not automatically extend to determining a strategy for draw, because beside ‘ drawing positions ’ — that are natural analogues of winning positions — a draw may occur by the repetition of a position that has already been visited. Nevertheless, one of the learning algorithms presented here can be modified to learn a strategy for draw, too. Along the way, the algorithm first tests whether there exists a winning strategy for the player in question; and if the answer is negative, then it runs for a drawing strategy.

Though the present study is purely mathematical, one of its main moti-

vations lies in the intersection of robotics and artificial intelligence where it is an important issue to automatize the process of learning.

Standard notation. In the sequel, $\mathbb{N}$ and $\mathbb{N}_0$ denote the set of positive and non-negative integers, respectively. A *partition* of a set M is a family $\{M_i \mid i \in I\}$ of pairwise disjoint non-empty subsets M_i of M such that $\bigcup_{i \in I} M_i = M$.

2 Terminal games

A *terminal game* $\mathcal{G}$ is defined as a 5-tuple $\mathcal{G} = (X, P, N, \mathcal{X}, \mathcal{W})$ with the following properties.

(G1) X and P are a non-empty sets;

(G2) N is a function from X into the power set 2^X of X;

(G3) $\mathcal{X} = \{X_p \mid p \in P\}$ is a partition of $X \setminus T$ where $T = \{x \in X \mid N(x) = \emptyset\}$;

(G4) $\mathcal{W} = \{W_p \mid p \in P\}$ is a family of subsets of T.

The elements of X, T, and P are called *positions*, *terminal positions*, and *players*, respectively. To illustrate with the NIM game, in this setting any size distribution of the piles belongs to *two* positions, distinguished by the player to turn.

The *graph* of the terminal game $\mathcal{G}$ is defined to be the directed graph $G = G(\mathcal{G})$ with vertex set $V(G) = X$ and edge set $E(G) = \{xy \mid y \in N(x)\}$. A position x' is *reachable* from a position x if there is a finite sequence $x_1, \ldots, x_k$ of positions such that $x_1 = x, x_k = x'$, and $x_{i+1} \in N(x_i)$ for every $i \in \{1, 2, \ldots, k-1\}$. Let $R(x)$ denote the set of all positions reachable from a position $x \in X$. A terminal game is called *locally finite* if $R(x)$ is finite for every position $x \in X$.

Let $Q \subseteq P$ be a set of players, and let $X_Q = \bigcup_{q \in Q} X_q$. A *$Q$-strategy* is a function f from X_Q into X such that $f(x) \in N(x)$ for every $x \in \bigcup_{q \in Q} X_q$. A P-strategy is called a *situation*. If s is a situation and $Q \subseteq P$ is a set of players, the restriction of s on X_Q is denoted by s_Q. Clearly, s_Q is a Q-strategy. If $Q = \{p\}$, we shall write p-strategy and s_p instead of $\{p\}$-strategy

and $s_{\{p\}}$, respectively. Furthermore, we shall write $-Q$ for the set $P \setminus Q$ and $-p$ for the set $P \setminus \{p\}$.

A *play* m of a terminal game $\mathcal{G} = (X, P, N, \mathcal{X}, \mathcal{W})$ is a pair $m = (x, s)$ such that $x \in X$ and s is a situation. The position x is called the *initial position* of the play $m = (x, s)$. The *trace* of a play $m = (x, s)$ is a sequence of positions $x_0, x_1, \ldots$ such that

(T1) $x_0 = x$;

(T2) $x_{i+1} = s(x_i)$ if $x_i \in X \setminus T$;

(T3) the trace is *finite* with *last element* x_k (for some $k \in \mathbb{N}$) if and only if either $x_k \in T$ or there is an index $0 \leq l < k$ such that $x_l = x_k$ and $x_0, \ldots, x_{k-1}$ are all distinct.

Let $X(m)$ denote the set of all positions that appear in the trace of m. Note that the trace of a play m is uniquely determined by m. A play is called *finite* if its trace is finite, and *infinite* otherwise. It follows from (T3) that a play m is infinite if and only if $X(m)$ is infinite. This implies that every play m of a locally finite terminal game is finite. Furthermore, it is not hard to see that if m is a play of a locally finite terminal game, then $X(m) \cap T \neq \emptyset$ if and only if no position appears more than once in the trace of m. If two plays m, m' have the same trace, we write $m \sim m'$.

A play m ends in a *draw* if and only if it is infinite or $X(m) \cap T = \emptyset$. Otherwise, the play is finite and the last element t of its trace is in T. Then, precisely the players $p \in P$ with $t \in W_p$ *win* the play and all other players *lose* it.

Intuitively, the idea behind the above definitions can be explained as follows. A terminal game $\mathcal{G}$ is played on the graph $G(\mathcal{G})$. At the beginning of a play $m = (x, s)$ a token is placed at the initial position x. Then the players move the token along the edges. If the token is at a position $y \in X_p$, player p moves the token from x to the position $s(y)$. The game ends if and only if the token is moved to a terminal position or it is placed at one and the same position for the second time. In the case that the token is moved to a terminal position $t \in T$, exactly those players p with $t \in W_p$ win and all other players lose the play. In all other cases the play ends in a draw.

3 Winning and drawing strategies

Let $\mathcal{G} = (X, N, P, \mathcal{X}, \mathcal{W})$ be a locally finite terminal game, $p \in P$ be a player and $x \in X$ a position. A p-strategy f is called a *winning strategy* for p at x if p wins every play $m = (x_0, s)$ with $x \in X(m)$ and $s_p = f$, and it is called a *drawing strategy* for p at x if p does not lose any play $m = (x_0, s)$ with $x \in X(m)$ and $s_p = f$. A position $x \in X$ is called a *winning position* for p if there is a winning strategy for p at x, and it is called a *drawing position* for p if there is a drawing strategy for p at x.

For the characterization of winning and drawing positions it is convenient to generalize these concepts as follows. Let $U \subseteq T$ and $Q \subseteq P$. A Q-strategy g is called a *U-forcing strategy* for Q at x if $X(m) \cap U \neq \emptyset$ for every play $m = (x_0, s)$ with $x \in X(m)$ and $s_Q = g$, and it is called a *U-avoiding strategy* for Q at x if $X(m) \cap U = \emptyset$ for every play $m = (x_0, s)$ with $x \in X(m)$ and $s_Q = g$. A position $x \in X$ is called a *U-forcing position* for Q if there is a U-forcing strategy for Q at x, and it is called a *U-avoiding position* for Q if there is a U-avoiding strategy for Q at x. Clearly, if x is a U-forcing position for Q, then it is not a U-avoiding position for $-Q$, and vice versa.

U-forcing and U-avoiding positions in terminal games can be characterized in a similar way as this can be done for winning positions in NIM-games. For $Q \subseteq P$ and $U \subseteq T$, we define a sequence $A_Q^0(U), A_Q^1(U), \ldots$ of subsets of X as follows.

(W1) $A_Q^0(U) = U$;

(W2) $A_Q^{i+1}(U) = A_Q^i(U) \cup \{x \in X_Q \mid N(x) \cap A_Q^i(U) \neq \emptyset\} \cup \{x \notin X_Q \mid N(x) \subseteq A_Q^i(U)\}$ for $i \in \mathbb{N}_0$.

Let $A_Q(U) = \bigcup\limits_{i \in \mathbb{N}_0} A_Q^i(U)$.

Proposition 3.1 *Let $m = (x_0, s)$ be a play, $k \in \mathbb{N}_0$, $U \subseteq T$, and $Q \subseteq P$.*

(a) If $X(m) \cap A_Q^k(U) \neq \emptyset$ and $s_Q(x) \in A_Q^i(U)$ for all $x \in X(m) \cap A_Q^{i+1}(U) \cap X_Q$ with $i \in \{0, 1, \ldots, k-1\}$, then $X(m) \cap U \neq \emptyset$.

(b) If $X(m) \setminus A_Q(U) \neq \emptyset$ and $s_{-Q}(x) \notin A_Q$ for all $x \in (X(m) \cap X_{-Q}) \setminus A_Q$, then $X(m) \cap U = \emptyset$.

(c) If $x \in A_Q(U)$, then x is a U-forcing position for Q.

(d) If $x \notin A_{-Q}(U)$, then x is a U-avoiding position for Q.

(e) If x is a U-forcing position for Q, then $x \in A_Q(U)$.

(f) If x is a U-avoiding position for Q, then $x \notin A_{-Q}(U)$.

Proof. Let $x_0, x_1, \ldots$ be the trace of m.

(a) Let j be the smallest integer such that $X(m) \cap A_Q^j(U) \neq \emptyset$, and let $x_l \in X(m) \cap A_Q^j(U)$. Suppose $j > 0$. Then $x_l \notin T$. If $x_l \in A_Q^j(U) \cap X_Q$, then it follows from the condition for s_Q that $x_{l+1} = s_Q(x_l) \in A_Q^{j-1}(U)$. Otherwise, $x_l \in A_j^Q(U) \setminus X_Q$, and (W2) implies that $x_{l+1} \in A_Q^{j-1}(U)$. In both cases, $x_{l+1} \in X(m) \cap A_Q^{j-1}(U)$, contradicting the minimality of j. Hence, $X(m) \cap A_Q^0(U) = X(m) \cap U \neq \emptyset$.

(b) Let l be the greatest integer such that $x_l \notin A_Q(U)$. If $x_l \in T$, then $X(m) \cap T = \{x_l\}$, and since $A_Q(U) \supseteq U$, $X(m) \cap U = \emptyset$. If $x_l \notin T$, we distinguish between two cases.

Case 1: $x_l \in X_Q \setminus A_Q(U)$. Then it follows from (W2) that $x_{l+1} \in X \setminus A_Q(U)$.

Case 2: $x_l \in X_{-Q} \setminus A_Q(U)$. Then the condition for s_{-Q} implies that $x_{l+1} = s_{-Q}(x_l) \notin A_Q(U)$.

By the maximality of l it follows that there is an index $l' < l$ such that $x_l = x_{l'}$. By (T3) this implies that $X(m) \cap U \subseteq X(m) \cap T = \emptyset$.

(c) follows from (a) and the fact that $N(x) \cap A_Q^i(U) \neq \emptyset$ for all $x \in A_Q^{i+1}(U) \cap X_Q$ with $i \in \{0, 1, \ldots, k-1\}$.

(d) follows from (b) and the fact that $N(x) \setminus A_Q \neq \emptyset$ for all for all $x \in X_{-Q} \setminus A_Q$.

(e) If $x \notin A_Q(U)$, then it follows from (d) that x is a U-avoiding position for $-Q$, and so x is not a U-forcing position for Q.

(f) If $x \in A_{-Q}(U)$, then it follows from (c) that x is a U-forcing position for $-Q$, and so x is not a U-avoiding position for Q. □

Clearly, a p-strategy f is a winning strategy for p at x if and only if it is a W_p-forcing strategy for p at x, and it is a drawing strategy for p at x if and only if it is a $(T \setminus W_p)$-avoiding strategy for p at x. Thus, we obtain the following corollary.

Corollary 3.2

(a) A position $x \in X$ is a winning position for a player $p \in P$ if and only if $x \in A_p(W_p)$.

(b) A position $x \in X$ is a drawing position for a player $p \in P$ if and only if $x \notin A_{-p}(T \setminus W_p)$.

4 Learning to win or to achieve a draw

4.1 Learning rules

Let $\mathcal{G} = (X, N, P, \mathcal{X}, \mathcal{W})$ be a locally finite terminal game, and consider an infinite sequence $m^1, m^2, \ldots$ of plays of $\mathcal{G}$ with $m^i = (x^i, s^i)$, where $i \in \mathbb{N}$. We say that player p applies a *deterministic k-learning rule* ($k \in \mathbb{N}$) in this sequence if his/her strategy s_p^i in the i-th play is uniquely determined by the traces of the preceding k plays $m^{i-k}, \ldots, m^{i-1}$ and his/her strategies $s_p^{i-k}, \ldots, s_p^{i-1}$ in these plays. (The strategies of p in the first k plays are arbitrary.)

In later subsections we will concentrate on a restricted type of deterministic 1-learning rules, where the decision of player p on the strategy to be applied in the next play depends only on the subsequence of the trace of the present play consisting of its positions in X_p and on the information whether p has won or lost the present play. We shall call such learning rules *independent 1-learning rules*.

It is clear that the learning success of a player applying an independent 1-learning rule also depends on how the other players play. In particular, if

player p applies an independent 1-learning rule in a sequence $m^1, m^2, \ldots$ of plays, then it cannot be guaranteed that for every winning position (resp., drawing position) x for p there is a number $L(x)$ such that player p will win (resp., will not lose) every play m^i with $x \in X(m^i)$ after having played at least $L(x)$ plays m^i with $x \in X(m^i)$. For example, p may win the first 10,000 plays without using a winning strategy, and none of the other players changes his strategy. Then, in the $10{,}001^{st}$ play some of the other players change their strategies and p loses this play.

A deterministic k-learning rule is called *W-effective* (resp., *D-effective*) if for every terminal game $\mathcal{G}$ and every winning position (resp., drawing position) x for a player p in $\mathcal{G}$ there is a number $N(\mathcal{G}, x)$ such that if p applies this learning rule in sequence $m^1, m^2, \ldots$ of plays of $\mathcal{G}$, then there are at most $N(\mathcal{G}, x)$ plays m^i with the following properties:

(i) $x \in X(m^i)$;

(ii) player p does not win (resp., loses) m^i.

We shall prove that there are W-effective and D-effective independent 1-learning rules. In case a player p applies a W-effective deterministic k-learning rule or a D-effective learning rule and all other players apply some (not necessarily effective) deterministic k-learning rules, then we can prove a stronger statement. We need the following proposition.

Proposition 4.1 *If all plays in the sequence $m^1, m^2, \ldots$ have the same initial position x, and every player applies some deterministic k-learning rule in this sequence, then there are positive integers r, t such that $m^i \sim m^{i+t}$ for every $i \geq r$.*

Proof. Since the terminal game $\mathcal{G}$ is locally finite, there are only finitely many distinct traces with initial position x. This implies that there are indices $0 \leq u < v$ such that $m^{u+j} \sim m^{v+j}$ for every $j \in \{0, \ldots, k-1\}$. Since every player applies a deterministic k-learning rule in this sequence, it follows that $m^i \sim m^{i+v-u}$ for every $i \geq u$. □

Theorem 4.2 *Let $\mathcal{G} = (X, N, P, \mathcal{X}, \mathcal{W})$ be a locally finite terminal game, $p \in P$ a player, and $k \in \mathbb{N}$. Then for every position $x \in A_p(W_p)$ (resp.,*

$x \notin A_{-p}(T \setminus W_p)$) there exists an integer $L(x,k)$ such that the following statement holds.

If $(m^1, m^2, \ldots)$ is a sequence of plays of $\mathcal{G}$ with initial position x, in which player p applies a W-effective (resp., D-effective) deterministic k-learning rule and every other player applies some k-learning rule, then player p wins every (does not lose any) play m^i with $i > L(x,k)$.

Proof. By Proposition 4.1, there are positive integers r, t such that $m^i \sim m^{i+t}$ for every $i \geq r$. This implies that p will win every (resp., cannot lose any) play m^i with $i \geq r$. For, if p does not win (resp., loses) a play m^i with $i \geq r$, then p would not win (resp., would lose) every play m^{i+kt} with $k \in \mathbb{N}_0$, contradicting the assumption that $N(\mathcal{G}, x)$ is finite. This shows that the theorem holds with $L(x,k) = r$. □

4.2 Latest Non-Winning Trace

Next, we define a special deterministic 1-learning rule. Let for every position $x \in X \setminus T$, π_x be a cyclic permutation of $N(x)$.

LNWT algorithm (Latest Non-Winning Trace). We say that a player $p \in P$ applies the *learning rule of Latest Non-Winning Trace* — LNWT, for short — in $m^1, m^2, \ldots$ if the strategy s_p^{i+1} of p in the play m^{i+1} is determined by s_p^i and the trace of the preceding play m^i as follows:

$$s_p^{i+1}(x) = \begin{cases} \pi_x(s_p^i(x)) & \text{if } p \text{ does not win } m^i \text{ and } x \in X(m^i) \\ s_p^i(x) & \text{otherwise} \end{cases}$$

for each $x \in X_p$ and $i \geq 1$.

Theorem 4.3 *Let $\mathcal{G} = (X, N, P, \mathcal{X}, \mathcal{W})$ be a locally finite terminal game, and $p \in P$ a player. Then, for every position $x \in A_p(W_p)$, there exists an integer $K(x)$ such that for every sequence $m^1, m^2, \ldots$ of plays of $\mathcal{G}$ in which player p applies the learning rule LNWT, there are at most $K(x)$ plays m^i with the following properties:*

(i) $x \in X(m^i)$;

(ii) player p does not win m^i.

Proof. If $x \in A_p(W_p)$, then there is a smallest integer n such that $x \in A_p^n(W_p)$. We shall prove the existence of $K(x)$ by induction on n.

By definition, p wins every play m^i with $X(m^i) \cap A_p^0(W_p) = X(m^i) \cap W_p \neq \emptyset$. Therefore, $K(x) = 0$ for every position $x \in A_p^0(W_p)$. Let $n \geq 1$, $x_0 \in A_p^n(W_p)$, and suppose that $K(x)$ exists for every position $x \in A_p^{n-1}(W_p)$. Let $m^{i_1}, m^{i_2}, \ldots$ be the subsequence of $m^1, m^2, \ldots$ consisting of all plays m^i with $x_0 \in X(m^i)$ and the property that p does not win m^i. Let l be length of the sequence $m^{i_1}, m^{i_2}, \ldots$, and let z_j denote the successor of x_0 in the trace of m^{i_j}. The inductive hypothesis implies that, for every $y \in A_p^{n-1}(W_p)$, there are at most $K(y)$ distinct indices $j \in \{1, \ldots, l\}$ such that $z_j = y$. We distinguish between two cases.

Case 1: $x_0 \in A_p^n(W_p) \cap X_p$. Then there is a position $y \in N(x_0) \cap A_p^{n-1}(W_p)$. Suppose that $l \geq (K(y)+1)|N(x_0)|$. Since p applies the learning rule LNWT, $z_{j+1} = \pi_{x_0}(z_j)$ for $j \in \{1, \ldots, l\}$. Consequently, there are $K(y)+1$ distinct indices $j \in \{1, \ldots, l\}$ with $z_j = y$ — a contradiction.

Case 2: $x_0 \in A_p^n(W_p) \setminus X_p$. By definition, $N(x_0) \subseteq A_p^{n-1}(W_p)$. Suppose that $l \geq 1 + \sum_{y \in N(x_0)} K(y)$. Then it follows from the pigeon-hole principle that there is a position $y \in A_p^{n-1}(W_p)$ such that $z_j = y$ for at least $K(y)+1$ distinct indices $j \in \{1, \ldots, l\}$ — a contradiction.

This shows that if $x_0 \in A_p^n(W_p) \cap X_p$, then $K(x_0) < (K(y)+1)|N(x_0)|$ for every position $y \in N(x_0) \cap A_p^{n-1}(W_p)$ (the worst case occurring if y is the last element of the list $N(x_0)$), and $K(x_0) \leq \sum_{y \in N(x_0)} K(y)$ otherwise. Consequently, $K(x)$ exists for all positions $x \in A_p(W_p)$. □

Corollary 4.4 *Let* $K_h = \max\{K(x) \mid x \in A_p^h(W_p)\}$. *If* $|N(x)| \leq d$ *for all* $x \in X$, *then* $K_h < (d+1)^h$.

Proof. It follows from the proof of Theorem 4.3 that K_h satisfies the inequality $K_{h+1} \leq (K_h + 1)d - 1$ for all $k \in \mathbb{N}$. This implies the claim, since $K_0 = 0$. □

The following example shows that the upper bound on $K(x)$ cannot be improved in general to subexponential in $|R(x)|$ (the number of positions

reachable from x) if a player applies LNWT. In fact, the number of plays m^k with $x \in X(m^k)$, lost by player p, will turn out to be exponential for all winning positions x in that game. For $n \in \mathbb{N}$ let $\mathcal{G}_n$ be the terminal game with just one player p and with the positions $a_1, \ldots, a_n; l_1, \ldots, l_n; w$ such that

- $N(a_i) = \{a_{i-1}, l_i\}$ for $i = 2, \ldots, n$ and $N(a_1) = \{w, l_1\}$;
- $l_1, \ldots, l_n$ and w are terminal positions;
- $W_p = \{w\}$.

It is clear that a_n (and every a_i) is a winning position for p, and the unique winning strategy s is $s(a_i) = a_{i-1}$ for $i > 1$ and $s(a_1) = w$. Let $m^1 = (a_n, s^1), m^2 = (a_n, s^2), \ldots$ be a sequence of plays of $\mathcal{G}_n$ in which p applies LNWT with the initial strategy s^1 given by $s^1(a_i) = l_i$ for $i = 1, \ldots, n$. Clearly, if p wins a play, then he/she will win all succeeding plays. Let $C(n)$ denote the number of plays lost in this sequence. Then p loses exactly the plays $m^1, \ldots, m^{C(n)}$. Obviously, p loses m^1 and $C(1) = 1$. Hence, we assume $n > 1$. By the definition of LNWT, $s_k(a_n)$ alternates between a_{n-1} and t_n in the subsequence of plays lost. If $s_k(a_n) = t_n$ (i.e., if k is odd), then p does not change his/her strategy for the subgame $\mathcal{G}_{n-1}$ after this play. If $s_k(a_n) = a_{n-1}$ (i.e., if k is even), then p changes his/her strategy also for the subgame $\mathcal{G}_{n-1}$ according to the learning rule LNWT. Since $s_{C(n)}(a_n) = s_1(a_n) = t_n$, this implies that $C(n) = 2C(n-1) + 1$. Finally, with $C(1) = 1$, we get $C(n) = 2^n - 1$.

4.3 Latest Non-Winning Position

An alternative way to learn a winning strategy if it exists is obtained if the player does not modify the strategy on the entire trace of the latest play lost, but only at a carefully chosen position of it. Instead of a circular list, here we assume that for each position $x \in X \setminus T$, $N(x)$ is stored as a *linear list* equipped with the *successor* operator σ_x. If y is the last element of $N(y)$, then $\sigma_x(y) := \mathsf{NIL}$ is defined to be a dummy symbol. Adopting this notation, the following learning method is proposed.

LNWP algorithm (Latest Non-Winning Position). We say that a player $p \in P$ applies the *learning rule of Latest Non-Winning Position* — LNWP, for short — in $m^1, m^2, \ldots$ if the strategy s_p^{i+1} of p in the play m^{i+1}

is determined by the preceding play m^i as follows:

$$s_p^{i+1}(x) = \begin{cases} \sigma_x(s_p^i(x)) & \text{if } p \text{ does not win } m^i \text{ and } x \text{ is the last position} \\ & \quad \text{in } X(m^i) \cap X_p \text{ with } \sigma_x(s_p^i(x)) \neq \mathsf{NIL} \\ s_p^i(x) & \text{otherwise} \end{cases}$$

for each $x \in X_p$ and $i \geq 1$. Before the first play, $s_p^1(x)$ is set to be the head of the list $N(x)$, for all $x \in X_p$.

We mention, that if the learning rule LNWP is applied to a game with just one player p, then it visits the positions in the same order as depth-first search until the first terminal vertex in W_p has been found.

Recall that $R(x)$ denotes the set of all positions reachable from $x \in X$. Let us define $M_p(x) = \sum_{y \in R(x) \cap X_p} |N(y)|$ for $p \in P$ and for $x \in X$.

Proposition 4.5 *Let $\mathcal{G} = (X, N, P, \mathcal{X}, \mathcal{W})$ be a locally finite terminal game, $p \in P$ a player, and let $m^1, m^2, \ldots$ be a sequence of plays of $\mathcal{G}$ in which player p applies the learning rule LNWP. Furthermore, let $x_0^i, x_1^i, \ldots$ be the trace of the play m^i for $i \in \mathbb{N}$, and let s_p^i be the strategy of p in m^i.*

(a) If $s_p^i(x) \in A_p^{k-1}(W_p)$ for a position $x \in A_p^k(W_p) \cap X_p$ (for any $k \in \mathbb{N}$), then $s_p^j(x) = s_p^i(x)$ for all $j \geq i$.

(b) If $X(m^i) \cap A_p(W_p) \neq \emptyset$, then p wins the play m^i if and only if $s_p^i = s_p^{i+1}$.

(c) There are at most $M_p(x)$ plays m^i such that

(i) $x \in X(m^i)$;

(ii) there is a $y \in R(x) \cap X_p$ with $s_p^i(y) \neq s_p^{i+1}(y)$.

Proof. (a) Suppose that the assertion is not true. Then there is a smallest index i_0 such that there exists a $k \in \mathbb{N}$ and a position $x \in A_p^k(W_p)$ with the property that $s_p^{i_0}(x) \in A_p^{k-1}(W_p)$ and $s_p^{i_0+1}(x) \neq s_p^{i_0}(x)$. It follows from the definition of the learning rule LNWP that $x \in X(m_0^i)$. Assume that $x = x_l^{i_0}$. Since player p applies LNWP and $s_p^{i_0+1}(x) \neq s_p^{i_0}(x)$, it follows that

(i) p does not win the play m^{i_0}, and

(ii) $y \notin X_p$ or $\sigma_y(s_p^{i_0}(y)) = \mathsf{NIL}$ for all $y = x_r^{i_0}$ with $r > l$.

Because of (i), and since $x_{l+1}^{i_0} = s_p^{i_0}(x) \in A_p^{k-1}(W_p)$, it follows from Proposition 3.1(a) that there is an index $t \geq l+1$ and a $q \in \mathbb{N}$ such that $x_t^{i_0} \in A_p^q(W_p)$ and $x_{t+1}^{i_0} = s_p^{i_0}(x_t^{i_0}) \notin A_p^{q-1}(W_p)$. Let $z = x_t^{i_0}$. We claim that $z \in A_p^q(W_p) \cap X_p$. For if $z \in (A_p^q(W_p) \setminus X_p)$, then, by the definition of $A_p^q(W_p)$, we have $N(z) \subseteq A_p^{q-1}(W_p)$ and consequently $x_{t+1}^{i_0} \in A_p^{q-1}(W_p)$. Hence, it follows from (ii) that $\sigma_z(s_p^{i_0}(z)) = \mathsf{NIL}$. Since $z \in A_p^q(W_p)$, there is an index $j < i_0$ such that $s_p^j(y) \in A_p^{q-1}(W_p)$ and $s_p^{j+1}(y) \notin A_p^{q-1}(W_p)$. This contradicts the minimality of i_0 and proves (a).

(b) It follows from the definition of the learning rule LNWP that if p wins the play m^i, then $s_p^i = s_p^{i+1}$. Conversely, if $s_p^i = s_p^{i+1}$ and p does not win the play m^i, then, again by LNWP, $\sigma_x(s_p^i(x)) = \mathsf{NIL}$ for all $x \in X(m^i) \cap X_p$. By (a) this implies that $\sigma_x(s_p^i(x)) \in A_p^{k-1}(W_p)$ for all $x \in X(m^i) \cap X_p \cap A_p^k(W_p)$ and all $k \in \mathbb{N}$. Since $X(m^i) \cap A_p(W_p) \neq \emptyset$, Proposition 3.1(a) implies that p wins the play m^i.

(c) By the definition of LNWP, for every $y \in X_p$ there are at most $|N(y)|$ plays m^i such that $s_p^i(y) \neq s_p^{i+1}(y)$. This implies (c). □

Theorem 4.6 *Let $\mathcal{G} = (X, N, P, \mathcal{X}, \mathcal{W})$ be a locally finite terminal game, $p \in P$ a player, $x \in X_p$, and $m^1, m^2, \ldots$ a sequence of plays of $\mathcal{G}$ in which player p applies the learning rule LNWP.*

(a) If $x \in A_p(W_p)$, then there are fewer than $M_p(x)$ plays m^i such that

(i) $x \in X(m^i)$;

(ii) p does not win m^i.

(b) If p does not win m^i and $\sigma_x(s_p^i(x)) = \mathsf{NIL}$ for all $x \in X(m^i) \cap X_p$, then $x \notin A_p(W_p)$.

Proof. (a) follows immediately from Proposition 4.5 (b) and (c).

(b) If $\sigma_x(s_p^i(x)) = \mathsf{NIL}$ for all $x \in X(m^i) \cap X_p$, then LNWP means $s_p^i = s_p^{i+1}$. Since p does not win m^i, Proposition 4.5(b) implies $x \notin A_p(W_p)$. □

If we restrict consideration to learning rules that change the strategy after the k-th play only in positions contained in $X_p \cap (\cup_{i=1}^{k} X(m^i))$, then the example at the end of Subsection 4.2 shows that in the worst case player p loses $\Omega(|X_p|)$ plays before a winning strategy is found. Hence, the learning rule LNWP is best possible in a sense. However, there are cases where LNWT works much faster than LNWP. For $n \in \mathbb{N}$ let $\mathcal{K}_n$ be the terminal game with just one player p and the positions $a, b, w, l_1, \ldots, l_n$, where $w, l_1, \ldots, l_n$ are terminal positions, $N(a) = \{w, b\}$, $N(b) = \{l_1, \ldots, l_n\}$, and $W_p = \{w\}$. A p-strategy s is a winning strategy for p at a if and only if $s(a) = w$. If p applies LNWP in a sequence of plays and the initial strategy is not a winning strategy, then p will lose exactly the first n plays. Applying LNWT, p will lose exactly one play.

4.4 Learning to achieve draw

We consider modifications, called LLT and LLP, of the learning rules LNWT and LNWP. Generally speaking, LLT and LLP are obtained from LNWT and LNWP, respectively, by modifying the rule from "if p does not win" to "if p loses". At first glance, it seems to be likely that these modified learning rules will find drawing strategies in the same way as the original rules find winning strategies. As we shall see, this is indeed true in case of the learning rule LLP, but it proves false for LLT.

Using the notation from Section 4, we define the learning rules LLT and LLP as follows.

LLT algorithm (Latest Losing Trace). We say that a player $p \in P$ applies the *learning rule of Latest Losing Trace* — LLT, for short — in $m^1, m^2, \ldots$ if the strategy s_p^{i+1} of p in the play m^{i+1} is determined by s_p^i and the trace of the preceding play m^i as follows:

$$s_p^{i+1}(x) = \begin{cases} \pi_x(s_p^i(x)) & \text{if } p \text{ loses } m^i \text{ and } x \in X(m^i) \\ s_p^i(x) & \text{otherwise} \end{cases}$$

for each $x \in X_p$ and $i \geq 1$.

LLP algorithm (Latest Losing Position). We say that a player $p \in P$ applies the *learning rule of Latest Losing Position* — LLP, for short — in

$m^1, m^2, \ldots$ if the strategy s_p^{i+1} of p in the play m^{i+1} is determined by the preceding play m^i as follows:

$$s_p^{i+1}(x) = \begin{cases} \sigma_x(s_p^i(x)) & \text{if } p \text{ loses } m^i \text{ and } x \text{ is the last position} \\ & \quad \text{in } X(m^i) \cap X_p \text{ with } \sigma_x(s_p^i(x)) \neq \mathsf{NIL} \\ s_p^i(x) & \text{otherwise} \end{cases}$$

for each $x \in X_p$ and $i \geq 1$. Before the first play, $s_p^1(x)$ is set to be the head of the list $N(x)$, for all $x \in X_p$.

We consider the following example. Let $\mathcal{H}$ be the terminal game with the only player p and the positions a, b, c, t such that

x	a	b	c	t
$N(x)$	$\{b, c\}$	$\{c, t\}$	$\{b, t\}$	$\emptyset$

and $W_p = \emptyset$. Clearly, t is the only terminal position, p has no winning strategy at any position, and there is a drawing strategy f for p at a with

x	a	b	c
$f(x)$	b	c	b

Let g and h be the following p-strategies:

x	a	b	c
$g(x)$	b	c	t
$h(x)$	c	t	b

The plays $m_g = (a, g)$ and $m_h = (a, h)$ have the traces a, b, c, t and a, c, b, t, respectively, and consequently, p loses m_g and m_h. Let $m^1 = (a, s^1), m^2 = (a, s^2), \ldots$ be an infinite sequence of plays of $\mathcal{H}$ in which player p applies the learning rule LLT where $s^1 = g$. (Note that, because of $|N(a)| = |N(b)| = |N(c)| = 2$, there is exactly one cyclic permutation π_x for any position $x \in \{a, b, c\}$.) It is not hard to see that $s^i = g$ if i is odd and $s^i = h$ otherwise. Hence, p loses all plays in this sequence. This shows that the learning rule LLT is not D-effective.

In case of the learning rule LLP we essentially follow the reasoning from Subsection 4.3. To simplify notation, we set $B_p = X \setminus A_{-p}(T \setminus W_p)$. Recall that a position x is a drawing position for a player $p \in P$ if and only

if $x \in B_p$ (see Corollary 3.2(b)). As in Subsection 4.3, we let $M_p(x) = \sum_{y \in R(x) \cap X_p} |N(y)|$ for $p \in P$ and for $x \in X$. We can now formulate the following analogue of Proposition 4.5.

Proposition 4.7 *Let $\mathcal{G} = (X, N, P, \mathcal{X}, \mathcal{W})$ be a locally finite terminal game, $p \in P$ a player, and let $m^1, m^2, \ldots$ be a sequence of plays of $\mathcal{G}$ in which player p applies the learning rule LLP. Furthermore, for $i \in \mathbb{N}$, let $x_0^i, x_1^i, \ldots$ be the trace of the play m^i and let s_p^i be the strategy of p in m^i.*

(a) If $s_p^i(x) \in B_p$ for a position $x \in B_p \cap X_p$, then $s_p^j(x) = s_p^i(x)$ for all $j \geq i$.

(b) If $X(m^i) \cap B_p \neq \emptyset$, then p wins the play m^i if and only if $s_p^i = s_p^{i+1}$.

(c) There are at most $M_p(x)$ plays m^i such that

(i) $x \in X(m^i)$;

(ii) there is a $y \in R(x) \cap X_p$ with $s_p^i(y) \neq s_p^{i+1}(y)$.

Proof. The argument is quite similar to the proof of Proposition 4.5.

(a) Suppose that the assertion is not true. Let i_0 be the smallest index such that there exists a position $x \in B_p$ with the property that $s_p^{i_0}(x) \in B_p$ and $s_p^{i_0+1}(x) \neq s_p^{i_0}(x)$. It follows from the definition of the learning rule LLP that $x \in X(m_0^i)$. Assume that $x = x_l^{i_0}$. Since player p applies LLP and $s_p^{i_0+1}(x) \neq s_p^{i_0}(x)$ it follows that

(i) p loses the play m^{i_0}, and

(ii) $y \notin X_p$ or $\sigma_y(s_p^{i_0}(y)) = \mathsf{NIL}$ for all $y = x_r^{i_0}$ with $r > l$.

Because of (i), and since $x_{l+1}^{i_0} = s_p^{i_0}(x) \in B_p$, it follows from Proposition 3.1(b) that there is an index $t \geq l+1$ such that $x_t^{i_0} \in B_p$ and $x_{t+1}^{i_0} = s_p^{i_0}(x_t^{i_0}) \notin B_p$. Let $z = x_t^{i_0}$. We claim that $z \in B_p \cap X_p$. For if $z \in B_p \setminus X_p = X_{-p} \setminus A_{-p}(T \setminus W_p)$, then, by the definition of $A_{-p}(T \setminus W_p)$, $N(z) \cap A_{-p}(T \setminus W_p) = \emptyset$, and consequently $x_{t+1}^{i_0} \notin A_{-p}(T \setminus W_p)$, i.e., $x_{t+1}^{i_0} \in B_p$. Hence, it follows from (ii) that $\sigma_z(s_p^{i_0}(z)) = \mathsf{NIL}$. Since $z \in B_p$, there is an index $j < i_0$ such that $s_p^j(y) \in B_p$ and $s_p^{j+1}(y) \notin B_p$. This contradicts the minimality of i_0 and proves (a).

(b) It follows from the definition of the learning rule LLP that if p does not lose the play m^i, then $s_p^i = s_p^{i+1}$. Conversely, if $s_p^i = s_p^{i+1}$ and p loses the play m^i, then LLP yields $\sigma_x(s_p^i(x)) = \mathsf{NIL}$ for all $x \in X(m^i) \cap X_p$. By (a) this implies that $\sigma_x(s_p^i(x)) \in B_p$ for all $x \in X(m^i) \cap X_p \cap B_p$. Since $X(m^i) \cap B_p \neq \emptyset$, Proposition 3.1(b) implies that p does not lose the play m^i.

(c) By the definition of LLP, for every $y \in X_p$ there are at most $|N(y)|$ plays m^i such that $s_p^i(y) \neq s_p^{i+1}(y)$. This implies (c). □

The analogue of Theorem 4.6 is

Theorem 4.8 *Let $\mathcal{G} = (X, N, P, \mathcal{X}, \mathcal{W})$ be a locally finite terminal game, $p \in P$ a player, $x \in X_p$ and $m^1, m^2, \ldots$ a sequence of plays of $\mathcal{G}$ in which player p applies the learning rule LLP.*

(a) If $x \in B_p$, then there are fewer than $M_p(x)$ plays m^i such that

(i) $x \in X(m^i)$;

(ii) p loses m^i.

(b) If p loses m^i and $\sigma_x(s_p^i(x)) = \mathsf{NIL}$ for all $x \in X(m^i) \cap X_p$, then $x \notin B_p$.

Proof. (a) follows immediately from Proposition 4.7 (b) and (c).

(b) If $\sigma_x(s_p^i(x)) = \mathsf{NIL}$ for all $x \in X(m^i) \cap X_p$, then LLP yields $s_p^i = s_p^{i+1}$. Since p loses m^i, Proposition 4.7(b) implies $x \notin B_p$. □

The learning rules LNWP and LLP can be combined in the following way. We consider a sequence $m^1, m^2, \ldots$ of plays with $m^k = (x, s^k)$ for $k \in \mathbb{N}$. It follows from Proposition 4.5(b) that if player p applies the learning rule LNWP and p does not win two consecutive plays using the same strategy, then x is not a winning position for p. If this situation occurs, then LNWP will never again change the strategy of p. Hence, it makes sense to use the learning rule LLP from now on in order to find a drawing strategy if there is one; i.e., we start again from the heads of all lists, and modify the rule from "if p does not win" to "if p loses". In this way, p can escape from losing an infinite number of plays, whenever possible.

5 Comments and discussion

This section includes some further comments.

(a) *Memory requirements.* The application of our learning rules LNWT, LNWP and LLP by a player p requires to maintain a table containing all positions $x \in X_p$ and the respective moves $s^k(x)$ used in the present play m^k. In case of the learning rule LNWT the positions in $X(m^k) \cap X_p$ must be marked. For the learning rules LNWP and LLP a pointer is needed that indicates the last element of the trace of m^k in X_p. Consequently, the table requires a memory of size $O(|X_p|)$. The storage of the cyclic permutations π_x or the successor operators σ_x requires a memory of size $O(\sum_{x \in X_p} |N(x)|)$. So, in total the memory requirements are linear in $\sum_{x \in X_p} |N(x)|$, i.e., linear in the number of edges of the game graph.

(b) *Reinforcement learning.* The learning rules discussed in the present paper belong, in some sense, to the theory of reinforcement learning. In the terminology of Sutton and Barto [1] those are evolutionary methods. LNWP and LLP are closely related to a trial-and-error learning system called MENACE (cf. [10, 11]). The learning rule LNWP can be described in a similar way. Each position is represented by a matchbox. Initially, these matchboxes contain for each possible move a colored bead whereas the moves are encoded by the colors. In the first play the player puts an arbitrary bead on top of the respective box. The colors of these beads represent his/her strategies. If the player wins, then he/she leaves his/her beads in place. Otherwise, he/she finds the last box used during play that contains yet another bead (if there is any), discards the bead on top of the box, and puts another bead out of the box on top of it.

(c) *Relative winning strategies.* We consider the case where one of the players, say q, in a 2-player game uses only a subset $\mathcal{S}$ of his/her possible strategies. It can occur that, for some fixed position x, the other player p has a strategy $f(g)$, called a *relative winning strategy*, such that p wins every play (x, s) with $s_q = g$ and $s_p = f(g)$ for every q-strategy $g \in \mathcal{S}$. Using a deterministic k-learning rule, player p has to decide on the strategy to be used in the next play before it starts. Since q may choose another $g' \in \mathcal{S}$ for which $f(g') \neq f(g)$, we obtain that k-learning rules do not suffice to learn relative winning strategies.

(d) *Randomized learning rules.* There is a natural randomized version rLNWT of the learning rule LNWT: Choose $s_p^{i+1}(x)$ uniformly at random if p does not win m^i and $x \in X(m^i)$, and let $s_p^{i+1}(x) = s_p^i(x)$ otherwise. If player p applies rLNWT, then s^k almost surely converges to a winning strategy (if there is any). Clearly, the learning rule LLT can be randomized in the same way. While LLT is not D-effective, the application of its randomization rLLT yields almost sure convergence to a drawing strategy if there is one (see [4]).

References

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

[2] C.L. Bouton, Nim, a game with a complete mathematical theory, Ann. Math. Princeton 3 (1901–1902), 35–39.

[3] E.R. Berlekamp, J.H. Conway, and R.K. Guy, *Winning Ways for Mathematical Games*, Academic Press, 1982.

[4] A. Ernst, *Gewinnstrategien auf Graphen*, Diploma-Thesis, Technische Univesrität Ilmenau, 2005.

[5] A.S. Fraenkel and O. Rahat, Infinite cyclic impartial games, Theoretical Computer Science 252 (2001), 13–22.

[6] A.S. Fraenkel, Combinatorial Games: Selected Bibliography with a Succinct Gourmet Introduction, The Electronic Journal of Combinatorics (http://www.combinatorics.org/Surveys/index.html)

[7] D. Fudenberg and J. Tirole, *Game Theory*, MIT Press, 1991.

[8] K. Jacobs, *Einführung in die Kombinatorik*, Walter de Gruyter, Berlin, New York, 1983.

[9] B. Kummer, *Spiele auf Graphen*, Deutscher Verlag der Wissenschaften, Berlin, 1979.

[10] D. Michie, Trial and error, In: S.A. Barnett and A. McLaren (Eds.), Science Survey, Part 2, pp. 129–145, Harmondsworth, Penguin, 1961.

[11] D. Michie, Experiments on the mechanisation of game learning. 1. Characterization of the model and its parameters, Computer Journal 1 (1963), 232–263.

[12] D. Michie, *On Machine Intelligence*, Edinburgh University Press, 1974.

[13] J. Yang, S. Liao and M. Pawlak, On a decomposition method for finding winning strategy in Hex game, in: Proceedings ADCOG: Internat. Conf. Application and Development of Computer Games (A. L. W. Sing, W. H. Man and W. Wai, eds.), City University of Honkong(2001), 96–111.