Записки научных
семинаров ПОМИ
Том 340, 2006 г.

Y. M. Lifshits, D. S. Pavlov

POTENTIAL THEORY FOR MEAN PAYOFF GAMES

Abstract. We present an $O(mn2^n \log Z)$ deterministic algorithm for solving the mean payoff game problem, m and n being respectively the number of arcs and vertices in the game graph and Z being the maximum weight (we assume that the weights are integer numbers). The theoretical basis for the algorithm is the *potential theory* for mean payoff games. This theory allows to restate the problem in terms of solving systems of algebraic equations with minima and maxima. Also we use *arc reweighting* technique to solve the mean payoff game problem by applying simple modifications to the game graph that do not change the set of winning strategies, obtaining at the end a trivial instance of the problem. We show that any game graph can be simplified by n reweightings.

1. Introduction

A *mean payoff game* is a game for two players, Alice and Bob, who move a token along the arcs of a weighted directed graph. Weights of all arcs are integer numbers. Roughly speaking, the first player wants to make the sums of the weights of the arcs passed so far tend to ∞ whereas the second player wants to make them tend to $-\infty$. Given the game graph G and the starting vertex u, the computational problem is to find the winner. The exact statement of the problem is given in the Section 2.

This problem has important applications to program verification. More precisely, the problem of model checking for modal μ-calculus reduces to the mean payoff game problem [12]. The mean payoff game problem is also interesting from theoretical viewpoint, since it is one of the few problems known to lie in NP and co-NP, but not known to lie in P so far.

In Section 3 we develop a potential theory for mean payoff games. The key idea is to assign to each vertex a pair of numbers called potentials. We give a game theoretical interpretation for these values and exhibit a system of equations that determines them in a unique way. Also we explain how to obtain a winning strategy from the system of potentials.

The first author was supported by the grants INTAS 04-77-7173 and NSh-8664.2006.1.

In Section 4 we present an $O(mn2^n \log Z)$ deterministic algorithm, which computes the system of potentials for a given graph, m and n being respectively the number of arcs and vertices in the game graph and Z being the maximum weight (all weights are integer numbers). The key idea is to use induction on vertices: Choose one vertex and remove it from the graph, compute the system of potentials for the reduced graph, and reinsert the removed vertex in the graph, computing both of its potentials using other potentials. Not all vertices can be removed in this way; if the process fails we try another vertex.

In Section 5 we describe a graph modification approach, which is based on the idea of *reweighting*. This idea was previously used to solve the assignment problem [13] and to find an optimal arborescence [3]. The key idea is to apply simple modifications to the game graph that do not change the set of winning strategies, obtaining at the end a trivial instance of the problem. We show that n reweightings are sufficient.

Related research. Mean payoff games were introduced in [4]. The survey [11] is an excellent introduction to the field. The idea of potentials goes back to Gallai [6]. A modern exposition can be found in [7]. Jurdziński introduced potentials for parity games in [2] while Björklund, Sandberg, and Vorobyov introduced potentials for mean payoff games in [2] and used them in their randomized $2^{O(\sqrt{n \log n})}$ algorithm. Our potentials are similar to theirs, although the details are different. A trivial deterministic $O(n^n)$ algorithm is a consequence of memoryless determinacy theorem proved in [4]. Several different pseudopolynomial algorithms are studied in [5, 15, 16].

2. Preliminaries

Definition. *A sink is a vertex without outgoing arcs. A mean payoff game is a triple (G, A, B), where G is a weighted directed graph without sinks. Weights of all arcs are integer numbers. Each vertex of the graph belongs to exactly one player: A is the set of Alice's vertices and B is the set of Bob's vertices.*

A position in the game is a vertex called the *current vertex*. At the beginning of the game the current vertex is set to some given vertex u. At each turn the owner of the current vertex chooses one outgoing arc of this vertex and moves the current vertex to the tip of this arc. The game is infinite, i.e., it does not terminate.

Alice's goal is to make the value $\liminf_{n\to\infty} z_n$ as large as possible whereas Bob's goal is to make the value $\limsup_{n\to\infty} z_n$ as small as possible. Here z_n is the average weight of the first n arcs passed during the game. We discuss later the exact rules that determine the winner.

Definition. *A positional strategy of Alice is a map f from the set A to the set of graph's arcs such that the arc $f(u)$ has u as its starting vertex. A positional strategy of Bob is defined in a similar way.*

Note that positional strategies are different from generic strategies, since a generic strategy depends on the current vertex and the history of the game (i.e., the arcs passed so far), whereas a positional strategy depends only on the current vertex.

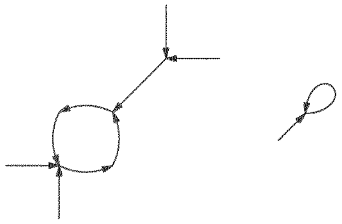
The following theorem proved in [1] states that the mean payoff game has a saddle point.

Determinacy theorem [1]. *For each vertex u, there is a number $v(u)$ called the game value for the vertex u such that Alice has a positional strategy allowing her to make* $\liminf_{n\to\infty} z_n \geqslant v(u)$ *whenever the game starts at the vertex u no matter what generic strategy Bob chooses and Bob has a positional strategy allowing him to make* $\limsup_{n\to\infty} z_n \leqslant v(u)$ *whenever the game starts at the vertex u no matter what generic strategy Alice chooses.*

Computational problems around mean payoff games. The initial computational problem is to find, for a given vertex u, the game value $v(u)$ and a winning positional strategy for each player as defined below. A simpler problem is to determine the set of vertices that satisfy the condition $v(u) > 0$, provided that the game graph does not have cycles of length 0. The length of a cycle is the sum of weights of its arcs. We now reduce the former problem to the latter in $\log Z$ steps, where Z is the maximum absolute value of arcs' weights. Recall that all weights are integers. Sections 3, 4, and 5 are devoted to the solution of the latter problem. Note that the algorithm for the reduced problem works with arbitrary real weights.

Theorem. *The game value for any vertex is a rational number with the denominator at most n and the absolute value at most Z.*

Proof. Consider a spanning subgraph H induced by a pair of positional winning strategies; each vertex of this graph has out-degree 1 and each connected component consists of a simple cycle with directed trees attached to it, like these:



It remains to note that the game values of all vertices of a connected component are equal to the average weight of the cycle of this component.

A simple but somewhat technical binary search completes the reduction. The necessary details are exhibited in [14]. The key idea is that two different rational numbers with denominators not exceeding n differ at least by $1/n(n-1)$. To determine whether $v(u) > r$ we subtract r from all weights and check whether $v(u) > 0$ in the new graph using our algorithm. A subtle technicality here is that we must guarantee the absence of simple cycles of length 0 in the new graph, since this is a necessary precondition for our algorithm. There are several ways to tackle this problem. The simplest is as follows. Since our algorithm works with arbitrary real weights, we can add a very small number $\epsilon > 0$ to all weights not changing the set of vertices with negative game values. In practice we do not add anything but whenever we encounter a sum of weights that equals 0 we treat it as a positive number. Another possibility is to always choose r in such a way that it is an irreducible fraction with denominator at least $n+1$.

For the rest of the paper we assume that the weights of the arcs are arbitrary real numbers and the graph does not contain simple cycles of length 0. The last statement is a necessary assumption for most of the theorems of the next section, since otherwise we have to deal with degenerate cases like ties.

We develop an algorithm for the following problem: Given a game graph without simple cycles of length 0 find all vertices with a positive game value. Note that in this problem the values of all vertices are either positive or negative, since absence of cycles of length 0 implies absence of vertices with value 0. Our algorithm uses only addition and comparison of weights.

Denote by y_n the sum of the weights of the first n arcs passed in the game. If $v(u) > 0$, then $\lim\limits_{n\to\infty} y_n = \infty$; otherwise $v(u) < 0$ and $\lim\limits_{n\to\infty} y_n = -\infty$. The converse is also true: If Alice chooses a generic strategy such that $\lim_{n\to\infty} y_n = \infty$ whatever generic strategy Bob chooses, then $v(u) \geqslant 0$ and, therefore, $v(u) > 0$; a similar assertion holds for Bob. Here we have used the fact that $v(u) \neq 0$.

We use the following formulation of the problem as the starting point for the theory developed in the next section: Two players move the token by the rules of the mean payoff game. Alice wins the game if $y_n \to \infty$, Bob wins the game if $y_n \to -\infty$. We proved in the previous paragraph that one of the players can win the game, i.e., there are no ties.

3. Potential theory for the mean payoff game

In this section we give exact definition of potentials, show how to obtain a winning set and a winning strategy for each player, and exhibit a system of equations such that its unique solution is the system of potentials. We give necessary and sufficient conditions needed to remove a vertex or an arc from the graph without changing the system of potentials.

Definition. *Suppose the game starts at the vertex u. Consider a generic strategy for Alice such that $y_n + p \geqslant 0$ for some number p no matter what generic strategy Bob chooses. The infimum of all such p through all strategies of Alice is called the potential of Alice at the vertex u and is denoted by $a(u)$. If no such p exists, we set $a(u) = \infty$. Consider a generic strategy for Bob such that $-y_n + p \geqslant 0$ for some number p no matter what generic strategy Alice chooses. The infimum of all such p through all strategies of Bob is called the potential of Bob at the vertex u and is denoted by $b(u)$. If no such p exists, we let $b(u) = \infty$.*

An ordered pair consisting of potentials of Alice and Bob is called the *system of potentials*.

By definition a potential is either a nonnegative real number or infinity.

Finiteness theorem. *Exactly one of the following conditions holds for each vertex u:*

- *The vertex u is winning for Alice, $0 \leqslant a(u) < \infty$, and $b(u) = \infty$.*
- *The vertex u is winning for Bob, $a(u) = \infty$, and $0 \leqslant b(u) < \infty$.*

Proof. Suppose that the vertex u is winning for Alice. Then she can make $\lim_{n\to\infty} y_n = \infty$; this implies that $b(u) = \infty$.

To prove that $a(u)$ is finite, choose any winning strategy for Alice and remove all arcs starting in Alice's vertex and not used in the strategy. The resulting graph does not have cycles of negative length that are reachable from the vertex u, since otherwise Bob can win the game. Thus we can assume that the game terminates when we encounter any vertex for the second time, since such operation does not change the minimum of y_n. Since there is only a finite number of such games, y_n is bounded from below for all strategies of Bob, therefore $a(u)$ is finite. The second case is proved by the same argument.

We proceed to the *game theoretical interpretation* of the potentials. Imagine that there is some initial amount r of prize money. The weights of the arcs passed so far are added to this amount. Thus w dollars are put on the balance if $w > 0$ or $-w$ dollars are taken from the balance if $w < 0$. The potential p of Alice at the vertex u in this interpretation is characterized by the following condition: If $r > p$, then Alice can play in such a way that the balance is always positive during the game; if $r < p$, then Bob can play in such a way that the balance becomes negative at least once during the game. Bob's potential q is characterized in a similar way: if we put $-r$ dollars on the balance at the beginning of the game, Bob can maintain negative balance whenever $r > q$; if however $r < q$, then Alice can make the balance positive at least once.

We prove later that Alice has a positional strategy that allows her to maintain nonnegative balance throughout the game if the balance at the beginning of the game is equal to her potential at the starting vertex. Such a strategy is called an *optimal* strategy.

Our definition of potentials was nonconstructive; now we exhibit explicit equations for the potentials.

Main theorem. *The following system of equations has exactly one solution. This solution is equal to the system of potentials.*

$$a(u) = \begin{cases} \min\max(0, a(v) - w), & \text{if the vertex } u \text{ belongs to Alice;} \\ \max\max(0, a(v) - w), & \text{if the vertex } u \text{ belongs to Bob.} \end{cases} \tag{1}$$

Here the outer minimum and maximum are taken over all outgoing arcs of the vertex u and w denotes the weight of the arc $u \to v$. The formulae for Bob potentials are the same except that the weights of the arcs are negated:

$$b(u) = \begin{cases} \min\max(0, b(v) + w), & \text{if the vertex } u \text{ belongs to Bob;} \\ \max\max(0, b(v) + w), & \text{if the vertex } u \text{ belongs to Alice.} \end{cases} \tag{2}$$

The last equation is a non-degeneracy condition:

$$\text{Exactly one of the numbers } a(u) \text{ and } b(u) \text{ is finite.} \tag{3}$$

Proof of existence. We prove that the potentials of Alice and Bob satisfy equations (1)–(3).

If the only outgoing arc of the vertex u is the arc $u \to v$ of weight w, then $a(u) = \max(0, a(v) - w)$.

Suppose the vertex u belongs to Alice and has several outgoing arcs. Denote by p the minimum of the value $\max(0, a(v) - w)$ over all outgoing arcs of the vertex u. Obviously, p dollars are enough to win the game that starts at the vertex u, since Alice can choose the arc that attains the minimum; hence $a(u) \leqslant p$. On the other hand, Alice needs at least p dollars to win the game that starts at the vertex u, since whatever arc she chooses at the first turn, $\max(0, a(v) - w) \geqslant p$; thus $p \leqslant a(u)$. We immediately obtain that $a(u) = p$. Using similar arguments when the vertex u belongs to Bob, we obtain equation (1).

Using similar arguments for Bob's potentials, we obtain equation (2).

Condition (3), which follows from Finiteness Theorem is needed to eliminate degenerate solutions; for example, we can set all $a(u)$ and $b(u)$ to ∞ without violating equations (1)–(2).

Proof of uniqueness. Suppose that $\bar{a}$ and $\bar{b}$ satisfy equations (1)–(3); we prove that $\bar{a}(u)$ and $\bar{b}(u)$ are the potentials of the vertex u. For the sake of being definite we assume that $\bar{a}(u) \neq \infty$ and $\bar{b}(u) = \infty$. The other case $\bar{a}(u) = \infty$ and $\bar{b}(u) \neq \infty$ is treated in a similar way. Note that formulae (1)–(2) guarantee that $\bar{a}(u)$ and $\bar{b}(u)$ are nonnegative.

Denote by f the balance of the game. Denote by v the current vertex. We prove that Alice can choose a positional strategy in such a way that the value of $f - \bar{a}(v)$ does not decrease during the game no matter what generic strategy Bob chooses. Moreover, Bob can choose a positional strategy in such a way that the same value does not increase during the game no matter what generic strategy Alice chooses until a vertex u such that $\bar{a}(u) = 0$ is encountered. Note that the statement is asymmetric with respect to Alice and Bob due to the last condition.

For her positional strategy Alice chooses arcs that give the minimum value in equation (1) whereas Bob chooses arcs that give the maximum value in the same equation.

First we prove the former statement. Suppose that Alice moves the token along the arc $u \to v$ of weight w. Then $\bar{a}(u) = \max(0, \bar{a}(v) - w)$ due to the choice of the strategy, therefore $\bar{a}(u) \geqslant \bar{a}(v) - w$. Hence $f + w - \bar{a}(v) \geqslant f - \bar{a}(u)$. If Bob moves the token along the arc $u \to v$ of weight w, then $\bar{a}(u) \geqslant \max(0, \bar{a}(v) - w) \geqslant \bar{a}(v) - w$ and the same argument applies.

The latter statement is proved in a similar way. Suppose that Alice moves the token along the arc $u \to v$ of weight w. We can assume that $\bar{a}(u) \neq 0$. Then $0 < \bar{a}(u) \leqslant \max(0, \bar{a}(v) - w) = \bar{a}(v) - w$, therefore $\bar{a}(u) \leqslant \bar{a}(v) - w$. Hence $f + w - \bar{a}(v) \leqslant f - \bar{a}(u)$. If Bob moves the token along the arc $u \to v$ of weight w, then $\bar{a}(u) = \max(0, \bar{a}(v) - w) = \bar{a}(v) - w$ due to the choice of the strategy and the same argument applies.

To complete the proof of uniqueness it remains to show the following. Suppose that the game starts at the vertex u with the balance r. If $r > \bar{a}(u)$ then Alice can choose her strategy in such a way that the balance always stays positive no matter what generic strategy Bob chooses. If $r < \bar{a}(u)$ then Bob can choose his strategy in such a way that the balance becomes negative at least once.

The first statement follows immediately from what we proved above, since Alice can choose her strategy in such a way that $f - \bar{a}(u)$ does not decrease and, therefore, stays positive during the game. Since $\bar{a}(u) \geqslant 0$, the balance is always positive.

To prove the second statement we note that Bob can choose his strategy in such a way that $f - \bar{a}(u)$ does not increase (and, therefore, stays negative) until a vertex v with $\bar{a}(v) = 0$ is encountered. If this is the case, then from $f - \bar{a}(v) < 0$ and $\bar{a}(v) = 0$ we infer that $f < 0$. If, however, we do not encounter a vertex v such that $a(v) = 0$ then the game lasts indefinitely and we encounter infinitely many simple cycles during the game. Each of these cycles has negative length, since the value of $f - \bar{a}(v)$ does not increase and the game graph does not have cycles of length 0. Since there is only a finite number of simple cycles in the graph, their length is bounded from above by a negative number. Hence $f - \bar{a}(v) \to -\infty$ and, therefore, $f \to -\infty$.

This theorem yields us an easy way to check whether a given set of values is the system of potentials. Indeed, it suffices to check whether equations (1)–(3) are satisfied.

Note that the theorem is wrong when we allow cycles of length 0. For example, if the game graph is a simple cycle of length 0, then adding the same value to all potentials of Alice and Bob does not affect the validity of equations (1)–(3).

Special properties of potentials. The arcs at which the finite minimum or the finite maximum in equations (1)–(2) is attained are called *hot* arcs. By definition, each vertex has at least one hot outgoing arc.

Restriction theorem. *Suppose H is a subgraph of the graph G. (A subgraph is obtained by removing some vertices and some arcs from the* *graph.) The system of potentials for the graph H is a restriction of the system of potentials for the graph G if and only if each vertex of H has at least one outgoing arc (belonging to H) that is hot in the graph G.*

Proof. Substituting the original potentials in equations (1)–(2) for the graph H yields true statements if and only if the minima and maxima stay the same. This is true if and only if each vertex of H has at least one outgoing arc that is hot in the graph G.

This theorem immediately yields a number of interesting consequences.

An arc is called *critical* if the system of potentials changes after removing this arc from the graph or if this arc is the only outgoing arc of its starting vertex.

Arc removing theorem. *An arc $u \to v$ is critical if and only if it is the only hot outgoing arc of the vertex u.*

For the sake of brevity the system of potentials for the subgraph induced by the set of vertices W is called the *system of potentials for the set W*.

Vertex restriction theorem. *Suppose W is a set of vertices such that each vertex in W has at least one hot outgoing arc ending in W; then the system of potentials for the set W is obtained by restricting the original system of potentials to the set W. The converse is also true.*

A vertex is called *critical* if at least one potential changes after removal of this vertex from the graph.

Vertex removing theorem. *A vertex is critical if and only if it has entering critical arcs apart from self-loops.*

Strategy theorem. *A positional strategy is optimal if and only if all its arcs are hot. Therefore, each player has an optimal strategy, which is winning for him on his winning set.*

Proof. Apply Restriction Theorem to the subgraph obtained by removing arcs that are not used in the strategy.

Zero theorem. *If $a(u) \neq \infty$ for some u, then $a(v) = 0$ for some v. The same statement holds for b.*

Proof. Without loss of generality assume that all potentials of Alice are finite, applying if necessary restriction theorem. Each vertex has at least one hot outgoing arc, hence we have at least one simple cycle consisting entirely of hot arcs. If all potentials of Alice are positive, the relation $a(u) = a(v) - w$ holds for each hot arc $u \to v$. Summing the last relation over all arcs of the cycle defined above we obtain that the length of this cycle is equal to 0. This leads us to a contradiction.

4. The algorithm

Informal ideas. The key idea of our algorithm is to compute the system of potentials for the graph G using the systems of potentials for the graphs of the form $G \setminus \{u\}$, where u is an arbitrary vertex of the graph.

Consider a spanning subgraph H induced by a fixed pair of optimal positional strategies; each vertex of this graph has out-degree 1 and each connected component consists of a cycle with directed trees attached to it. (This graph is used to justify the algorithm and is not constructed during the actual computation.) Suppose we remove from the graph G a vertex u of in-degree 0 in H. By Vertex Removing Theorem the potentials of the other vertices do not change. We can use equations (1)–(2) to compute the potential of the vertex u if the potentials of all other vertices are known. Hence we need not know whether the removed vertex u has in-degree 0, since at the time of reinsertion of the vertex u in the graph we compute its potentials and check whether the resulting set of values is the potential system. Therefore, we try each vertex in turn and see whether we succeed with it.

If we try each vertex and do not obtain a feasible solution, then H consists entirely of cycles. By Zero Theorem at least one of the potentials of Alice or Bob is equal to 0. For the sake of being definite we assume that $a(u) = 0$; this implies that $b(u) = \infty$. If we replace all outgoing arcs of the vertex u by a self-loop of weight 1, then the system of potentials does not change; this operation, however, breaks one cycle. Again, we need not know whether $a(u) = 0$; after computing the system of potentials for the modified graph we check whether these potentials are suitable for the original graph; in case of a failure we try another vertex in place of u.

In most of the cases we proceed with removing a vertex of in-degree 0 in H. The only exception to this rule is a situation when the vertex u is isolated in the graph H. In this case we remove this vertex, compute the system of potentials for the remaining graph, and reinsert u by the method described above. If the reinsertion process fails, we try another vertex in place of u.

Formal description of the algorithm. Suppose that all outgoing arcs of the vertex u are removed and a new self-loop of weight 1 (respectively -1) is added to the vertex u; then the system of potentials for the set W is called the *system of potentials for the set W with the vertex u marked for Alice (respectively Bob)*.

Now we describe our algorithm in a more detailed and formalized way. For each set of vertices W we compute the system of potentials for the set W possibly with one of its vertices marked for one of the players; this is done in the following way:

(0) If $W = \emptyset$, then the system of potentials is empty.

(1) To compute the system of potentials for the set W with vertex u marked for the player X do the following.

   (a) At this step we assume that the vertex u is isolated. Set the potential of the player X at the vertex u to 0, the potential of its opponent at the same vertex to ∞, and the potentials of the other vertices to the potentials for the set $W \setminus \{u\}$. If the resulting set of values is the system of potentials, store it; otherwise go to step (b).

   (b) At this step we assume that the vertex u is not isolated. For each vertex $v \neq u$ do the following. Set the potential of all vertices except v to the potentials for the set $W \setminus \{v\}$ with the vertex u marked for the player X. Compute the potentials of the vertex v using formulae (1)–(2). If the resulting set of values is the system of potentials, store it and end the loop; otherwise proceed with the next vertex v. At least one vertex v has in-degree 0 in the graph H; this vertex is noncritical and can be removed from the graph, thereby giving us at least one solution.

(2) To compute the system of potentials for the set W we iterate over all vertices u, and check whether the system of potentials for the set W with the vertex u marked for Alice or Bob is the system of potentials for the original graph. At least one of these systems is feasible.

Careful implementation of this approach yields worst case running time $O(mn2^n)$, where m is the number of the arcs and n is the number of the vertices.

5. Arc reweighting

In this section we describe another approach for solving the mean payoff game problem. The key idea is to modify the graph by a sequence of simple operations that do not change the set of winning strategies, obtaining at the end a trivial instance of the problem. The same approach is used in the Hungarian algorithm for the assignment problem [13] and in an algorithm for constructing the shortest arborescence [3].

The basic operation is *arc reweighting*, which is defined in the following way: add d to the weights of all outgoing arcs of the vertex u and subtract d from the weights of all entering arcs of this vertex. Here u and d are the parameters of the operation.

Note that arc reweighting does not change the set of winning strategies, although it may change the system of potentials.

Denote the winning sets of Alice and Bob by U and V. The game graph is called *trivial* if the following conditions are satisfied:

- At least one outgoing arc of any Alice's vertex in U is nonnegative and ends in U.
- All outgoing arcs of any Bob's vertex in U are nonnegative and end in U.
- At least one outgoing arc of any Bob's vertex in V is nonpositive and ends in V.
- All outgoing arcs of any Alice's vertex in V are nonpositive and end in V.

Note that all potentials of a trivial graph are equal to 0 or ∞; this follows from equations (1)–(2). If we know the winning sets, then we can compute the system of potentials easily, since $a(u) = 0$ if and only if the vertex u is winning for Alice.

Theorem. *We can recognize trivial graphs and find their winning sets in $O(m)$ time, where m is the number of the arcs in the graph.*

Proof. Suppose that the vertex u belongs to Alice and has at least one positive outgoing arc. Then $u \in U$. On the other hand, if all outgoing arcs of the vertex u are negative, then $u \in V$; otherwise we can remove all negative outgoing arcs of the vertex u from the graph, since Alice cannot use them anyway in her optimal strategy. Thus we assume that all outgoing arcs of Alice's vertex that is not assigned to the sets U and V have weight 0. In the same way we assign several Bob's vertices to the sets U and V, making the weights of all outgoing arcs of the other vertices equal to 0.

Now perform a depth-first search using the following rules. If the vertex u has at least one outgoing arc ending in the winning set of its owner, then this vertex belongs to his winning set; if all outgoing arcs of the vertex u end in the winning set of the other player, then this vertex belongs to his winning set.

Suppose we cannot apply this rule any more; denote by W the set of the remaining vertices. Any arc starting in W has weight 0. The subgraph induced by the set W is acyclic and has a sink if $W \neq \emptyset$. This sink was classified as winning for one of the players by the depth-first search. Therefore, $W = \emptyset$. At the end of the algorithm we check the validity of the resulting system of potentials and if it is not valid, then the graph is not trivial.

Theorem. *There exists a sequence of reweightings turning a given graph into a trivial one.*

Proof. Apply arc reweighting to each vertex u with the parameter $a(u)$ if $a(u) \neq \infty$ or with the parameter $-b(u)$ if $b(u) \neq \infty$. Equations (1)–(3) are easily verified.

Note that n simple operations are enough to reduce our problem to the trivial case.

6. Conclusion and open problems

We defined the system of potentials, showed how to obtain a winning set and a winning strategy for each player from the system of potentials, and exhibited a system of equations such that its unique solution is the system of potentials.

We constructed an $O(mn2^n)$ deterministic algorithm, which computes the system of potentials for a given graph. The idea of the algorithm is to compute the system of potentials for the whole graph from the system of potentials for one of its subgraphs. This algorithm yields an $O(mn2^n \log Z)$ deterministic algorithm for the mean payoff game problem.

We introduced a reweighting operation, which modifies the game graph preserving the set of winning strategies. We proved the existence of a short sequence of reweightings that reduced a given graph to a trivial graph. The proof was non-constructive.

Our algorithm shows that induction on vertices is applicable to the mean payoff game problem, thus generalizing in some sense the approach suggested in [10].

Potentials seem to play a crucial rôle in most of the current approaches to the problem, hence it is important to understand them well. We showed different ways such as game theoretical interpretation, algebraic min-max equations, and reweightings leading to one and the same notion.

Open problems. The main open problem in the area is to construct a polynomial time algorithm for the mean payoff game problem. Since the problem lies in NP and co-NP it is likely that such an algorithm exists. An intermediate step to this goal is to develop an $O(2^{\sqrt{n}})$ deterministic algorithm.

We conjecture that there is a polynomial Turing reduction from the problem of potential computing to the problem of computing winning sets. In other words, can we compute potentials effectively if we can compute the winning sets effectively?

We conjecture that there is an effective way to find a sequence of reweightings that reduces a given graph to the trivial case, like in [13, 3]. We ask for new invariant operations, more general than reweighting.

References

1. Henrik Björklund, Sven Sandberg, and Sergei Vorobyov, *Memoryless determinacy of parity and mean payoff games: a simple proof.* — Theoretical Computer Science **310** (2004), 365–378.
2. Henrik Björklund, Sven Sandberg, and Sergei Vorobyov, *A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games.* — DIMACS Technical Report **5** (2004).
3. Y. J. Chu and T. H. Liu, *On the shortest arborescence of a directed graph.* — Science Sinica **14** (1965), 1396–1400.
4. A. Ehrenfeucht and J. Mycielski, *Positional strategies for mean payoff games.* — International Journal of Game Theory **8** (1979), 109–113.
5. V. A. Gurvich, A. V. Karzanov, and L. G. Khachiyan, *Cyclic games and an algorithm to find minimax cycle means in directed graphs.* — USSR Computational Mathematics and Mathematical Physics **28** (1988), 85–91.
6. T. Gallai, *Maximum-Minimum Sätze über Graphen.* — Acta Mathematica Academiae Scientiarum Hungaricae **9** (1958), 395–434.
7. Alexander Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency.* Springer-Verlag, Berlin (2003).
8. Donald B. Johnson, *Efficient algorithms for shortest paths in sparse networks.* — Journal of the ACM **24** (1977), 1–13.
9. Marcin Jurdziński, *Small progress measures for solving parity games*, in: *Proceedings of* 17*th Annual Symposium on Theoretical Aspects of Computer Science*, Hotst Reichel and Sophie Tison, editors (Springer, 2000), 290–301. (Lecture Notes in Computer Science, volume 1770.)
10. Marcin Jurdziński, Mike Paterson, and Uri Zwick, *A Deterministic Subexponential Algorithm for Solving Parity Games.* — to appear.
11. Hartmut Klauck, *Algorithms for Parity Games*, in: *Automata, Logics, and Infinite Games*, E. Grädel et al., editors (Springer-Verlag, 2002), 107–129. (Lecture Notes in Computer Science, volume 2500.)
12. Dexter Kozen, *Results on the propositional μ-calculus.* — Theoretical Computer Science **27** (1983), 333–354.
13. H. W. Kuhn, *The hungarian method for the assignment problem.* — Naval Research Logistics Quarterly **2** (1955), 83–97.

14. Stephen Kwek and Kurt Mehlhorn, *Optimal Search for Rationals.* — Information Processing Letters **86** (2003), 23–26.
15. N. Pisaruk, *Mean cost cyclical games.* — Mathematics of Operations Research **24** (1999), 817–828.
16. Uri Zwick and Mike Paterson, *The complexity of mean payoff games on graphs.* — Theoretical Computer Science **158** (1996), 343–359.

С.-Петербургское отделение
Математического института
им. В. А. Стеклова РАН

E-mail: yura@logic.pdmi.ras.ru

Поступило 4 марта 2006 г.

С.-Петербургский
институт точной механики и оптики

E-mail: pavlov239@gmail.com