Algorithmica (2007) OF1–OF14
DOI: 10.1007/s00453-007-0175-3

Algorithmica

Simple Stochastic Games, Parity Games, Mean Payoff Games and Discounted Payoff Games Are All LP-Type Problems[1]

Nir Halman[2]

Abstract. We show that a Simple Stochastic Game (SSG) can be formulated as an LP-type problem. Using this formulation, and the known algorithm of Sharir and Welzl [SW] for LP-type problems, we obtain the first strongly subexponential solution for SSGs (a strongly subexponential algorithm has only been known for binary SSGs [L]). Using known reductions between various games, we achieve the first strongly subexponential solutions for Discounted and Mean Payoff Games. We also give alternative simple proofs for the best known upper bounds for Parity Games and binary SSGs.

To the best of our knowledge, the LP-type framework has been used so far only in order to yield linear or close to linear time algorithms for various problems in computational geometry and location theory. Our approach demonstrates the applicability of the LP-type framework in other fields, and for achieving subexponential algorithms.

Key Words. Simple stochastic games, Subexponential randomized algorithms, LP-type framework.

1. Introduction. Sharir and Welzl [SW] defined a model which generalizes Linear Programming (LP) and called it the LP-type model (see definitions in Section 2.2). An LP-type problem of combinatorial dimension d, where d is independent of the size n of the problem, is called *fixed dimensional*. Several algorithms that solve LP-type problems in time linear in n are known, such as the ones of Sharir and Welzl [SW] or Kalai [Ka]. The $O(n)$ time algorithm of Clarkson [Cl], which was originally formulated to solve LP, fits the LP-type model as well [CM], [GW1]. By formulating problems as *fixed-dimensional* LP-type problems, and using the LP-type algorithms, one can obtain linear time algorithms to various optimization problems, mainly in computational geometry and location theory, as shown in [A] and [MSW].

The algorithms of [Ka] and [SW] run in time subexponential in d. In this paper we use the LP-type framework in order to give the first strongly subexponential solution for Simple Stochastic Games, Discounted Payoff Games and Mean Payoff Games (defined below). To the best of our knowledge, this is the first application of the LP-type framework for solving a problem which is neither in computational geometry nor in location theory. Moreover, it is the first application of variable-dimensional LP-type problems.

A *Simple Stochastic Game* (SSG) is defined on a directed graph with three types of vertices, *min*, *max* and *average*, along with two sink vertices, the 0-*sink* and the 1-*sink*.

[1] This work is based upon a chapter in the author's Ph.D. dissertation, prepared at Tel Aviv University under the supervision of Professor Arie Tamir.
[2] Massachusetts Institute of Technology, Cambridge, MA 02139, USA. halman@mit.edu.

Received March 14, 2006.

The sink vertices have no outgoing edges. For every average vertex a_k, the outgoing edges from a_k have positive rational weights such that the sum of their weights is 1. The outgoing edges from the min and max vertices are unweighted. One of the vertices is a *start* vertex.

The game is a contest between two players, 0 and 1. It is played in the following way. Begin by placing a token on the start vertex. When the token is on a min vertex y_j, player 0 moves it along one of the outgoing edges of y_j. When the token is on a max vertex x_i, player 1 moves it along one of the outgoing edges of x_i. When the token is on an average vertex a_k, the edge along which the token is moved is determined randomly, in proportion to the weights of the edges outgoing from a_k. The game ends when one of the sink vertices is reached. The goal of player 1 is to reach the 1-sink. The goal of player 0 is to avoid the 1-sink.

Before the game begins, player 1 chooses an outgoing edge from each max vertex. These selected edges will define a *strategy* for player 1. During the game, whenever the token is on a max vertex, player 1 will move the token along the edge that is included in his strategy. (Defining strategies deterministically in this way does not result in any loss of generality [Co].) Similarly, a choice of an outgoing edge for each min vertex is a strategy for player 0. Given an SSG, we would like to find the "optimal strategies" for both players (we formally define optimal strategies in Section 2.1). The corresponding *decision problem* is to determine whether player 1 wins with probability greater than $\frac{1}{2}$ when both players use their optimal strategies. SSGs are closely connected with the development of algorithms for automatic verification ("model-checking") and synthesis of hardware and software systems [GW2].

A *binary* SSG is a special case of an SSG, where the outgoing degrees of the min and max vertices are bounded by 2, and where all average vertices have exactly two outgoing edges of weight $\frac{1}{2}$ each. Zwick and Paterson [ZP] gave a simple polynomial reduction from an SSG with n vertices and n_e edges, in which the denominators of all the (rational) probabilities are at most W, to a binary SSG with $n' = O((n + n_e) \log W)$ vertices and $n'_e = O(n_e \log W)$ edges. We note that n_e may be quadratic in n, so the number of vertices in the binary game resulting by the reduction, n', may be $\Theta(n^2)$. In this case the running time of the best known algorithm for binary SSGs (which is $e^{O(\sqrt{n'})}$ time) becomes exponential in n. This is the reason why the proof of Corollary 7.4 in [GW2] is in error.

Condon [Co] was the first to study SSGs from a complexity theory point of view. She showed that the SSG decision problem is in NP $\cap$ co-NP (and even in UP $\cap$ co-UP [J1]), and hence is unlikely to be NP-complete, but at the same time it is not known to be in P, despite substantial effort (see [EJS], [L], [ZP], [Se], [BCJ$^+$], [J2], [GW2], and [BSV1]). Some exponential algorithms for SSGs are described in [MC]. The first subexponential algorithm for binary SSGs was an $e^{O(\sqrt{n})}$ time ad hoc algorithm obtained in 1995 by Ludwig [L], for games with n vertices.

We show (see also [Ha1]) the first subexponential solution for (non-binary) SSGs that halt with probability 1 (and consequently for the SSG decision problem, see definitions below) where the game given consists of n nodes and n_e edges, in which the denominators of all the (rational) probabilities are at most W. The idea is to formulate the SSG as an LP-type problem, and then to calculate optimal strategies for both players by the LP-type algorithm of [SW]. Independently, Björklund et al. [BSV1] developed several ad hoc subexponential algorithms for this problem. Their approach is different. They first formulate the objective function of this problem as a special function (which they call either the RLG, CLG or CU function). They then "adapt" the algorithm of [L], as well as the LP-type subexponential algorithms of [SW] and [Ka], to solve these functions. The algorithms of [Ha1] and [BSV1] are randomized and run in $e^{O(\sqrt{n \log n})} \times C(n, n_e, W)$ time, where $C(x, y, z)$ is the time needed to solve a linear program with x variables, y constraints and z being the (binary) coding size of the input numbers. The best known algorithms for solving variable-dimensional LP problems (e.g., the one of Khachiyan [Kh]) perform a polynomial number of operations in x, y and z. Since the number of operations they perform depends on the size of the input numbers, they are *not* strongly polynomial. This implies that the algorithms of [Ha1] and [BSV1] are *not* strongly subexponential. In this paper we obtain the first *strongly* $e^{O(\sqrt{n \log n})}$ subexponential solution for SSG. This algorithm is faster than the previous algorithms when W is much greater than n.

SSGs are a restriction of stochastic games introduced by Shapley [Sh], some 50 years ago. Many variants of SSGs have been studied since then (see [PV] for a survey). In this work we consider three variants of SSGs: Parity Games (PGs), Mean Payoff Games (MPGs) and Discounted Payoff Games (DPGs).

A *Parity Game* is defined on a directed graph with two types of vertices, 0 and 1. Each vertex has a positive integer *color* and has at least one outgoing edge. (The number of colors k may be as big as the number of vertices.) One of the vertices is a *start* vertex. Similarly to SSG, the game is a contest between two players, 0 and 1. It is played in the following way. Begin by placing a token on the start vertex. When the token is on a 0 (1) vertex, player 0 (1) moves it along one of its outgoing edges, respectively. The players construct an infinite path called a *play*. The largest vertex color j occurring infinitely often in a play determines the winner. Player 0 wins if j is even and player 1 wins if j is odd.

Other variants of SSGs are MPGs and DPGs. Each of these games is an infinite two-person game played on a directed graph $G = (V, E)$ in which each vertex has at least one edge going out of it. Let $\omega\colon\ E \rightarrow \{-W, \ldots, 0, \ldots, W\}$ be a function that assigns an integral weight to each edge of G. One of the vertices, say a_0, is a *start* vertex. The first player chooses an edge $e_1 = (a_0, a_1) \in E$. The second player then chooses an edge $e_2 = (a_1, a_2) \in E$, and so on. The first player wants to maximize the function f_1 while the second player wants to minimize f_2. In MPGs

$$f_1 = \lim \inf_{t \to \infty} \frac{1}{t} \sum_{i=1}^{t} \omega(e_i); \qquad f_2 = \lim \sup_{t \to \infty} \frac{1}{t} \sum_{i=1}^{t} \omega(e_i).$$

In DPGs we are also given a rational *discounting factor* λ with $0 < \lambda < 1$. The weight of the ith edge, e_i, chosen by the players is now multiplied by $(1 - \lambda)\lambda^i$ and

$$f_1 = f_2 = (1 - \lambda) \sum_{i=1}^{\infty} \lambda^i \omega(e_i).$$

SSGs are a *generalization* of PGs, MPGs and DPGs in the sense that there exists a polynomial time reduction from them to (non-binary) SSGs that halt with probability 1.

Table 1.

Problem	Previous results	Our results
Binary SSG	Strongly $e^{O(\sqrt{n})}$ [L]	Strongly $e^{O(\sqrt{n})}$
SSG	$e^{O(\sqrt{n \log n})} \times C(n, n_e, W)$ [Ha1, BSV1]	Strongly $e^{O(\sqrt{n \log n})}$
PG	Strongly $e^{O(\sqrt{n \log n})}$ [BSV2]	Strongly $e^{O(\sqrt{n \log n})}$
MPG	$\log W \cdot e^{O(\sqrt{n \log n})}$ [BSV3]	Strongly $e^{O(\sqrt{n \log n})}$
DPG	No subexponential algorithm is known	Strongly $e^{O(\sqrt{n \log n})}$

(See definition in Section 2.2. See [Pu] or Lemma 7.5 in [GW2] for a reduction of PG to MPG, and [ZP] for a reduction of MPG and DPG to SSG.) The decision problems corresponding to PGs, MPGs and DPGs are also known to be in NP ∩ co-NP. No polynomial time algorithm for any of these decision problems is yet known. Using the above reductions we get that our strongly subexponential $e^{O(\sqrt{n \log n})}$ time algorithm for SSGs solves PGs, MPGs and DPGs in the same (strongly) subexponential time bound.

Independently of our results, Björklund et al. [BSV2], [BSV3] developed ad hoc strongly $e^{O(\sqrt{n \log n})}$ algorithms for PGs and for the decision problem corresponding to MPGs (they also provided a $\log W \cdot e^{O(\sqrt{n \log n})}$ algorithm for MPGs). We summarize the previous best results and our results in Table 1.

As seen in the table, our algorithmic results improve upon the best known algorithms for SSGs, MPGs and DPGs. We also give alternative simple proofs for the best known upper bounds for PGs and binary SSGs. While Ludwig [L] and Björkland et al. [BSV1]–[BSV3] developed ad hoc algorithms for each of the specific games they solved, we use only one *unifying* algorithm for all of these five games—the LP-type algorithm of Sharis and Welzl [SW].

2. Definitions and Previous Results.

In this section we review the definitions and the results known about SSGs and LP-type problems.

2.1. *Simple Stochastic Games*.

Although this section is self-contained, it is strongly based upon the first two sections in [L].

Let $G = (V = X \uplus N \uplus A \uplus \{v_0, v_1\}, E = S \uplus D \uplus AA)$ be an SSG. v_0 is the 0-*sink* and v_1 is the 1-*sink*. $X = \{x_1, x_2, \dots, x_d\}$, $N = \{y_1, y_2, \dots, y_m\}$ and $A = \{a_1, \dots, a_p\}$ are the sets of *max*, *min* and *average* vertices, respectively. Let $n = |V| = d + m + p + 2$ be the number of vertices in the graph. For $i = 1, \dots, d$ ($i = 1, \dots, m$; $i = 1, \dots, p$) let X_i (N_i, A_i) be the set of edges outgoing from the max (min, average) vertex x_i (y_i, a_i), respectively. Let $S = \biguplus_{i=1}^{d} X_i$ ($D = \biguplus_{i=1}^{m} N_i$; $AA = \biguplus_{i=1}^{p} A_i$) be the set of the edges outgoing from vertices of X (N, A), respectively. For every $i = 1, \dots, p$, every edge $e \in A_i$ is given a positive rational weight $pr(e) < 1$ such that $\sum_{e \in A_i} pr(e) = 1$.

Formally, a strategy for player 1 is a function $\sigma\colon \{1, 2, \dots, d\} \to V$ which indicates for every vertex $x_i \in X$, an outgoing edge $(x_i, \sigma(i)) \in X_i$ that player 1 will choose. We define a strategy τ for player 0 similarly. Let σ, τ be a pair of strategies for players 1 and 0. Construct the graph $G_{\sigma,\tau} = (V, S_\sigma \cup D_\tau \cup AA)$ where $S_\sigma = \{(x_i, \sigma(i)) \mid i = 1, \dots, d\}$

and $D_\tau = \{(y_i, \tau(i)) \mid i = 1, \ldots, m\}$. An SSG *halts with probability* 1 if and only if for *all* pairs of strategies σ, τ, every vertex in $G_{\sigma,\tau}$ has a path to a sink vertex. In the results that follow, we restrict our discussion to games that halt with probability 1 (Lemma 2.2 below shows that this restriction can be applied to the decision problem without loss of generality).

We define the *value* of a vertex $z \in V$ with respect to a pair of strategies σ, τ, denoted by $v_{\sigma,\tau}(z)$, to be the probability that player 1 will win the game if the start vertex is z and the players use strategies σ and τ.

We say that a vertex $x_i \in X$ is *stable* with respect to a pair of strategies σ, τ if $v_{\sigma,\tau}(x_i) = \max\{v_{\sigma,\tau}(z) \mid (x_i, z) \in E\}$. Similarly, we say that a vertex $y_i \in N$ is stable with respect to a pair of strategies σ, τ if $v_{\sigma,\tau}(y_i) = \min\{v_{\sigma,\tau}(z) \mid (y_i, z) \in E\}$. We say that a vertex is *unstable* if it is not stable.

Let σ, τ be a pair of strategies for players 1 and 0. The strategy τ is said to be *optimal with respect to* σ if every min vertex is stable with respect to σ, τ. We will let $\tau(\sigma)$ denote an optimal strategy for player 0 with respect to the player 1 strategy σ. Optimal strategies for player 1 are defined similarly. σ, τ are said to be *optimal* if each strategy is optimal with respect to the other.

The lemmas below were originally stated for binary SSGs. Using the polynomial reduction of [ZP] from SSGs to binary SSGs, they are valid also for SSGs.

LEMMA 2.1 (Lemma 2 in [L]). *Let $G = (V, E)$ be an SSG that halts with probability* 1. *Then there is a pair of optimal strategies σ^*, τ^* for players* 1 *and* 0 *for the game G.*

The *value* of an SSG G is the value of the start vertex with respect to a pair of optimal strategies for the two players. For every vertex $z \in V$ we denote by $v(z)$ the value of the SSG G where the start vertex is z.

LEMMA 2.2 (Lemma 3 in [L]). *Given an SSG G, we can construct a new game G' in time polynomial in the size of G such that G' has the same number of min and max vertices as G, the value of G' is greater than $\frac{1}{2}$ if and only if the value of G is greater than $\frac{1}{2}$, and G' halts with probability* 1.

The next two lemmas show that the definition of "optimal" is suitable in the sense that an optimal strategy for either player optimizes the value of every vertex from that player's point of view.

LEMMA 2.3 (Lemma 4 in [L]). *Let $G = (V, E)$ be an SSG that halts with probability* 1, *and let σ^*, τ^* be a pair of optimal strategies. Then for all $z \in V$,*

$$v_{\sigma^*,\tau^*}(z) = \max_{\sigma} \min_{\tau} v_{\sigma,\tau}(z).$$

LEMMA 2.4 (Lemma 5 in [L]). *Let $G = (V, E)$ be an SSG that halts with probability* 1. *Then for any vertex $z \in V$,*

$$\min_{\tau} \max_{\sigma} v_{\sigma,\tau}(z) = \max_{\sigma} \min_{\tau} v_{\sigma,\tau}(z).$$

So all pairs of optimal strategies give the same probability for player 1 to win.

If the game consists only of max and average vertices (i.e., $N = \emptyset$) or only of min and average vertices (i.e., $X = \emptyset$), then Derman [D] showed that a linear program can be constructed such that there is a one-to-one correspondence between basic feasible solutions and strategies. However, no such construction is known when the game consists of all three types of vertices. In the next section we show that an LP-type problem can be constructed such that there is a one-to-one correspondence between feasible solutions of the LP-type problem and strategies in the corresponding SSG.

LEMMA 2.5 [D]. *Let H be an SSG with no max vertices that halts with probability* 1. *Let n denote the number of vertices in H, let the vertices of H be labeled such that $V = \{1, 2, \ldots, n\}$, and let the* 0*-sink and the* 1*-sink be labeled $n-1$ and n, respectively. Then the optimal strategy for player* 0 (*with respect to the trivial player* 1 *strategy*) *can be found by solving the following linear program*:

$$
\begin{aligned}
&\textit{maximize} \sum_{i=1}^{n} v(i) \\
&\textit{subject to} \quad v(i) \le v(j) && \textit{if} \quad i \in N \quad \textit{and} \quad (i, j) \in E, \\
(1) \qquad & \qquad\qquad v(i) = \sum_{(i,j)\in E} pr(i, j)v(j) && \textit{if} \quad i \in A, \\
& \qquad\qquad v(n-1) = 0, \\
& \qquad\qquad v(n) = 1.
\end{aligned}
$$

We observe that the condition that H halts with probability 1 ensures the boundness of (1): for each $i \in V$, $v(i) \le 1$. Having the solution of the linear program (1) on hand, i.e., the value of the vertices in the graph, we find an optimal strategy τ for player 0 in the following way. For every $i = 1, \ldots, m$, $\tau(i) = j$ where $(i, j) \in E$ and $v(i) = v(j)$.

In Section 3 we use the following lemma for showing that the SSG can be formulated as an LP-type problem.

LEMMA 2.6 (Adaptation of Lemma 6 in [L]). *Let $G = (V, E)$ be an SSG that halts with probability* 1, *and let σ be a strategy for player* 1 *that is not optimal. Let $x_i \in X$ be a vertex that is unstable with respect to $\sigma, \tau(\sigma)$. Let σ' be a strategy that is obtained from σ by changing the strategy at vertex x_i such that $\forall j \ne i$, $\sigma'(j) = \sigma(j)$, $(x_i, \sigma'(i)) \in E$ and $v_{\sigma,\tau(\sigma)}(\sigma'(i)) = \max_{(x_i,z)\in E} v_{\sigma,\tau(\sigma)}(z)$. Then for all $z \in V$, $v_{\sigma',\tau(\sigma')}(z) \ge v_{\sigma,\tau(\sigma)}(z)$, and for some $z \in V$, $v_{\sigma',\tau(\sigma')}(z) > v_{\sigma,\tau(\sigma)}(z)$.*

Proofs of Lemmas 2.1–2.4 and 2.6 can be found in [Co] (most of these are based on proofs by Shapley [Sh], Howard [Ho] and Derman [D]). We conclude this section by stating a lemma proved in [L].

LEMMA 2.7 (Lemma 9 in [L]). *Any function $f \colon \mathbb{N} \to \mathbb{N}$ satisfying*

$$f(d) \le f(d-1) + \frac{1}{d} \sum_{i=1}^{d-1} f(i) + 1$$

for $d > 1$ *and*

$$f(1) \leq 1,$$

has $f(d) \leq e^{2\sqrt{d-1}}$, *for all* $d \geq 1$.

2.2. *LP-Type Problems*. Most of the definitions in this subsection are taken from [MSW] and [A]. We start by defining a general class of problems:

DEFINITION 2.8. An *abstract problem* is a tuple (H, ω) where H is a finite set of elements (which we call *constraints*) and ω is an *objective function* from 2^H to some totally ordered set Λ which contains a special maximal (minimal) element ∞ $(-\infty)$, respectively. The goal is to compute $\omega(H)$.

DEFINITION 2.9. An *LP-type problem* is an abstract problem (H, ω) that obeys the following conditions (when we write $<, \leq, =$, etc., we mean under the ordered set Λ).

Monotonicity: For all $F \subseteq G \subseteq H$, $\omega(F) \leq \omega(G)$.

Locality: For all $F \subseteq G \subseteq H$ such that $\omega(F) = \omega(G) \neq -\infty$ and for each $h \in H$, if $\omega(G \cup \{h\}) > \omega(G)$ then $\omega(F \cup \{h\}) > \omega(F)$.

Let $G \subseteq H$ be arbitrary. If $\omega(G) = \infty$, we say G is *infeasible*; otherwise we call G *feasible*. If $\omega(G) = -\infty$, we say G is *unbounded*; otherwise we call G *bounded*. A *basis* B is a set $B \subseteq H$ with $\omega(B') < \omega(B)$ for all proper subsets B' of B. A *basis for* G is a basis with $\omega(B) = \omega(G)$. The *combinatorial dimension* d of an LP-type problem is the maximum size of any basis for any feasible sub-family G. An LP-type problem of combinatorial dimension d is called d-*dimensional*. An LP-type problem is *fixed dimensional* if d is a constant, i.e., independent of the size n of the problem.

An LP-type algorithm takes a d-dimensional LP-type problem (H, ω) and returns a basis B for H. The randomized algorithm of [SW] (which was re-analyzed in [MSW]), Function lptype (see Figure 1), gets as an input the set of constraints H, and a candidate basis $C \subseteq H$. C is not necessarily a basis for H. It can be viewed as some auxiliary information one gets for the computation of the solution which has no influence on the output of the procedure (but it influences its efficiency). The algorithm uses two primitive operations. A *basis computation* takes a family G of at most $d + 1$ constraints and finds

Function lptype(H, C)

1. **if** $H = C$ **then return** C
2. **else**
   (a) choose at random $h \in H \backslash C$
   (b) B ←lptype($H \backslash \{h\}$, C)
   (c) **if** Violation(B, h) **then return** lptype(H, Basis($B \cup \{h\}$))
   (d) **else return** B

Fig. 1. The algorithm of [SW] returns a basis for the LP-type problem on H, when given a candidate basis C.

a basis for G. A *violation test* takes a basis B and a constraint h, and returns false if and only if B is a basis for $B \cup \{h\}$.

Let (H, ω) be a d-dimensional LP-type problem and let $n = |H|$. Let t_b (t_v) be the time required for a basis computation (a violation test), respectively. Let n_b (n_v) be the number of basis computations (violation tests) performed throughout the execution of the algorithm. Matoušek et al. [MSW] show that $n_v \le n_b n$, which implies a crude upper bound of $O(n_b(t_v n + t_b))$ for the running time of their algorithm. When the size of every basis for any feasible subfamily is exactly d, and $d \le n \le \sqrt{d}e^{d/4}$, they [MSW] show that $n_b = e^{O(\sqrt{d \ln d})}$, so the algorithm of [SW] runs in randomized $O(e^{O(\sqrt{d \ln d})}(t_v n + t_b))$ time, i.e., subexponential in dimension d of the problem. If n is large we use the algorithm of [Cl], which takes an LP-type problem of a large size n and breaks it into several smaller problems of size $9d^2$. These small problems we solve by the algorithm of [SW]. The overall (randomized) running time of the combined algorithm is $O(e^{O(\sqrt{d \ln d})}(t_v n + t_b \log n))$. To the best of our knowledge, the LP-type framework is used to solve only problems in computational geometry and location theory, and in linear or close to linear time (see [A], [MSW] and the references therein).

3. Formulating the SSG as an LP-type Problem. Let $G = (V = X \cup N \cup A \cup \{v_0, v_1\}, E = S \cup D \cup AA)$ be an SSG. For every $S' \subseteq S$ such that every max vertex $x \in X$ has an outgoing edge in S' (i.e., $X_i \cap S' \neq \emptyset, \forall i \in \{1, \dots, d\}$) we say that $G(S') = (V, S' \cup D \cup AA)$ is a simple stochastic *sub-game* of G (with respect to edges outgoing from max vertices). We note that every non-sink vertex in $G(S')$ has at least one outgoing edge. We also note that if G halts with probability 1 then so does $G(S')$.

Let $\Lambda = \mathbb{R} \cup \{\infty, -\infty\}$ be a set where ∞ ($-\infty$) is a special maximal (minimal) element, respectively. Let $\sigma(S'), \tau(S')$ be an arbitrary pair of optimal strategies in $G(S')$. Similarly to [L], we define ω: $2^S \to \Lambda$ in the following way:

$$(2) \qquad \omega(S') = \begin{cases} -\infty & \text{if} \quad \exists i \in \{1, \dots, d\} \quad \text{s.t.} \quad X_i \cap S' = \emptyset, \\ \sum_{z \in V} v_{\sigma(S'), \tau(S')}(z) & \text{otherwise.} \end{cases}$$

We note that, due to Lemma 2.3, $\omega(S')$ is well defined since for any $z \in V$ the value of z with respect to any pair of optimal strategies is the same. We also note that S' is always feasible, and it is bounded if and only if $G(S')$ is a sub-game of G.

We note that if σ, τ is a pair of strategies in the game G, and since it is not always true that $S_\sigma \subseteq S'$, then σ, τ is not necessarily a pair of strategies in $G(S')$. We also note that if σ, τ is a pair of optimal strategies in $G(S')$, it is a pair of strategies in G which is not necessarily optimal since the set of possible strategies for player 1 in G contains the one in $G(S')$ (an example of such G, $G(S')$, σ and τ can be easily constructed). We observe that there is a sufficient condition that ensures the optimality of σ, τ in G.

OBSERVATION 3.1. *Let $G = (V = X \cup N \cup A \cup \{v_0, v_1\}, S \cup D \cup AA)$ be an SSG that halts with probability 1. Let $G(S') = (X \cup N \cup A \cup \{v_0, v_1\}, S' \cup D \cup AA)$ be a simple stochastic sub-game of G. Let σ, τ be a pair of optimal strategies in $G(S')$. If $\omega(S) = \omega(S')$ then σ, τ is a pair of optimal strategies in G as well.*

PROOF. We first note that since the sets of outgoing edges from min vertices in G and $G(S')$ are identical (and equal to D), τ is an optimal strategy in G with respect to σ, so $\tau = \tau(\sigma)$.

We now claim that σ is an optimal strategy in G with respect to τ. Suppose on the contrary that σ is not an optimal strategy in G with respect to τ. By the non-optimality of σ in G there is a vertex $x_i \in X$ that is unstable with respect to $\sigma, \tau(\sigma)$. (That is, $v_{\sigma,\tau(\sigma)}(x_i) \neq \max_{(x_i,z)\in S} v_{\sigma,\tau(\sigma)}(z)$.) Let σ' be a strategy that is obtained from σ by changing the strategy at vertex x_i such that $\forall j \neq i$, $\sigma'(j) = \sigma(j)$, $(x_i, \sigma'(i)) \in S$ and $v_{\sigma,\tau(\sigma)}(\sigma'(i)) = \max_{(x_i,z)\in S} v_{\sigma,\tau(\sigma)}(z)$. Due to Lemma 2.6

$$(3) \quad \forall z \in V, \qquad v_{\sigma',\tau(\sigma')}(z) \geq v_{\sigma,\tau(\sigma)}(z) \quad \text{and} \quad \exists z \in V, \qquad v_{\sigma',\tau(\sigma')}(z) > v_{\sigma,\tau(\sigma)}(z).$$

If σ', $\tau(\sigma')$ is a pair of optimal strategies in G we get that $\omega(S) > \omega(S')$ in contradiction. Otherwise let σ^*, τ^* be an arbitrary pair of optimal strategies in G. Let $\mathcal{S} = \{\sigma\colon X \to V \mid \forall x_i \in X\ (x_i, \sigma(i)) \in S\}$ ($\mathcal{T} = \{\tau\colon N \to V \mid \forall y_i \in N\ (y_i, \tau(i)) \in D\}$) be the set of strategies for player 1 (player 0) in game G, respectively. By Lemma 2.3 we get that for all $z \in V$,

$$(4) \qquad v_{\sigma^*,\tau^*}(z) = \max_{\sigma\in\mathcal{S}} \min_{\tau\in\mathcal{T}} v_{\sigma,\tau}(z),$$

and by choosing the specific strategy $\sigma' \in \mathcal{S}$ we obtain

$$(5) \qquad \max_{\sigma\in\mathcal{S}} \min_{\tau\in\mathcal{T}} v_{\sigma,\tau}(z) \geq \min_{\tau\in\mathcal{T}} v_{\sigma',\tau}(z) = v_{\sigma',\tau(\sigma')}(z).$$

Combining (4) and (5) together implies that $v_{\sigma^*,\tau^*}(z) \geq v_{\sigma',\tau(\sigma')}(z)$, so from (3) we get again that $\omega(S) > \omega(S')$ in contradiction. □

We are now ready to show

LEMMA 3.2. *An SSG* $(V = X \cup N \cup A \cup \{v_0, v_1\}, S \cup D \cup AA)$ *that halts with probability* 1 *is a* d*-dimensional LP-type problem* (S, ω) *where* ω *is as defined in* (2) *and* $d = |X|$.

PROOF. We consider the abstract problem (S, ω). The aim of player 1 is to maximize his chances to win (i.e., maximize ω). We need to show that the Monotonicity and Locality Conditions are met. Directly from the definitions of $\sigma(S')$, $\tau(S')$ and $\omega(S')$ $(S' \subseteq S)$ we get that the Monotonicity Condition is satisfied. (Adding more edges outgoing from vertices of the set X enlarges the set of possible strategies for player 1, and hence does not decrease the probability of player 1 to win.)

Now we show that the Locality Condition is met, that is, for all $S' \subseteq S$, and for all $S'' \subset S'$ such that $\omega(S') = \omega(S'') \neq -\infty$ and for each $e = (x_i, v) \in S$, if $\omega(S' \cup \{e\}) > \omega(S')$ then $\omega(S'' \cup \{e\}) > \omega(S'')$. Let σ, τ be a pair of optimal strategies in $G(S'')$. By Observation 3.1, σ, τ is also a pair of optimal strategies in $G(S')$, so all vertices in the game $G(S')$ are stable with respect to the pair of optimal strategies σ, τ. If $\omega(S' \cup \{e\}) > \omega(S')$ then σ, τ is not a pair of optimal strategies in $G(S' \cup \{e\})$, and there is at least one vertex in V which is unstable in $G(S' \cup \{e\})$. Since all the vertices in $G(S')$ are stable, the only vertex in $G(S' \cup \{e\})$ which is unstable is x_i (x_i is the only vertex in V which has different sets of outgoing edges in $G(S')$ and in $G(S' \cup \{e\})$). In this way (and since the vertices in $G(S' \cup \{e\})$ and $G(S'' \cup \{e\})$ have the same values with respect to σ, τ) x_i is unstable in $G(S'' \cup \{e\})$ as well. Due to Lemmas 2.6 and 2.3, as explained at the end of the proof of Observation 3.1, we get that $\omega(S'' \cup \{e\}) > \omega(S'')$ as needed. Considering the combinatorial dimension of the problem, we note that a basis B (in the LP-type sense) for any bounded $S' \subseteq S$ is the set of d edges corresponding to an optimal strategy σ, for player 1 in the sub-game $G(S')$ (i.e., $B = S_\sigma$). Hence the corresponding LP-type problem is d-dimensional. □

We define sub-games with respect to edges outgoing from min vertices. Let $G = (V = X \cup N \cup A \cup \{v_0, v_1\}, E = S \cup D \cup AA)$ be an SSG. For every $D' \subseteq D$ with $N_i \cap D' \neq \emptyset$, $\forall i \in \{1, \ldots, m\}$ we say that $G(D') = (V, S \cup D' \cup AA)$ is a simple stochastic *sub-game* of G. We let $\Lambda = \mathbb{R} \cup \{\infty, -\infty\}$ be as before, and let $\sigma(D'), \tau(D')$ be an arbitrary pair of optimal strategies in $G(D')$. We define $\nu\colon 2^D \to \Lambda$ in the following way:

$$\nu(D') = \begin{cases} -\infty & \text{if} \quad \exists i \in \{1, \ldots, m\} \quad \text{s.t.} \quad N_i \cap D' = \emptyset, \\ -\sum_{z \in V} v_{\sigma(D'),\tau(D')}(z) & \text{otherwise.} \end{cases} \tag{6}$$

We prove the following lemma in a similar way to which we proved Lemma 3.2.

LEMMA 3.3. *An SSG* $(V = X \cup N \cup A \cup \{v_0, v_1\}, S \cup D \cup AA)$ *that halts with probability* 1 *is an* m*-dimensional LP-type problem* (D, ν) *where* ν *is as defined in* (6) *and* $m = |D|$.

4. Solving SSGs, PGs, MPGs and DPGs in Strongly $e^{O(\sqrt{n \log n})}$ Time

THEOREM 4.1. *An SSG* $G = (V, S \cup D \cup AA)$ *that halts with probability* 1 *is solvable in strongly* $e^{O(\sqrt{n \log n})}$ *expected time*.

PROOF. We solve the SSG by calling Function lptype(H, C) of [SW] (see Figure 1) with $H = S$ and C as follows. We choose C to consist of d arbitrary edges from S, each one going out of a distinct vertex in V, so C is bounded. Note that all subsets of S visited throughout the execution of the algorithm are bounded as well. By Lemma 3.2 the SSG, $G(S) = (V, S \cup D \cup AA)$, is a d-dimensional LP-type problem (S, ω), so the algorithm correctly solves the problem. As explained in Section 2.2, Function lptype runs in $O(n_b(t_v|S| + t_b))$ expected time where $n_b = e^{O(\sqrt{d \log d})}$ is the number of basis computations, t_v is the time needed to perform a violation test and t_b is the time needed to perform a basis computation.

A basis for any feasible $S' \subseteq S$ is a set $B'_S = S_\sigma$ where σ is an optimal strategy with respect to $\tau(\sigma)$ for the sub-game $G(S')$. An edge $e = (x_i, z)$ violates B'_S (Violation(B'_S, e) = true) if and only if $v_{\sigma,\tau}(x_i) \neq \max\{v_{\sigma,\tau}(z), v_{\sigma,\tau}(\sigma(i))\}$. This can be checked in constant time, so $t_v = O(1)$.

Let σ' be the strategy obtained from σ by changing the strategy at vertex x_i (i.e., $\forall i \neq j, \sigma'(j) = \sigma(j)$ and $\sigma'(i) = z$). A basis computation for $B'_S \cup \{e\}$ (Basis$(B'_S \cup \{e\})$) is done by deciding which strategy amongst σ, σ' is optimal in the game $(V, B'_S \cup \{e\} \cup D \cup AA)$.

In order to decide this we need to find an optimal strategy of player 0 with respect to σ', i.e., to find an optimal strategy of player 0 in the sub-game $G(S_{\sigma'})$. We note that the outdegree of each max vertex in this sub-game is 1. By Lemma 3.3, $G(S_{\sigma'})$ is an m-dimensional LP-type problem (D, ν). We solve this problem by calling again the LP-type algorithm of [SW], in $O(n'_{\mathrm{b}}(t'_{\mathrm{v}}|D|+t'_{\mathrm{b}}))$ expected time where $n'_{\mathrm{b}} = e^{O(\sqrt{m \log m})}$. It is sufficient to show that t'_{v} and t'_{b} are both strongly polynomial in n. As before, $t'_{\mathrm{v}} = O(1)$. Let $B'_D = D_\tau$ be a basis for any bounded $D' \subseteq D$, and let $e = (y_i, z) \in D$ be an edge which violates B'_D. Let τ' be the strategy obtained from τ by changing the strategy at vertex y_i. A basis computation for $B'_D \cup \{e\}$ (Basis$(B'_D \cup \{e\})$) is done by deciding which strategy amongst τ, τ' is optimal in the game $(V, S_{\sigma'} \cup B'_D \cup \{e\} \cup AA)$. In order to decide this we need to find an optimal strategy of player 1 with respect to τ', i.e., to find an optimal strategy of player 1 in the sub-game $G(S_{\sigma'}, D_{\tau'})$. In this sub-game the outdegree of each min and max vertex is 1, so only one trivial strategy exists for each of the players. The values of this game for each of its vertices can be computed by solving a system of n linear equations with n variables. This can be done for instance in strongly $O(n^3)$ time by Gaussian elimination, so t'_{b} is as needed. □

Due to Lemma 2.2 we get:

COROLLARY 4.2. *The decision problem corresponding to SSGs is solvable in strongly* $e^{O(\sqrt{n \log n})}$ *expected time.*

Since DPGs, MPGs and PGs are all strongly polynomially reducible to SSGs that halt with probability 1 we get

COROLLARY 4.3. *DPGs, MPGs and PGs are all solvable in strongly* $e^{O(\sqrt{n \log n})}$ *expected time.*

5. Solving Binary SSGs in Strongly $e^{O(\sqrt{n})}$ Time. As reviewed in Section 2.2, the running time of Function lptype is linearly dependent on $n_{\mathrm{b}}(t_{\mathrm{v}}n + t_{\mathrm{b}})$. In the previous section we used the result of [MSW] that $n_{\mathrm{b}} = e^{O(\sqrt{n \log n})}$. We now show that for binary SSGs this bound can be lowered to $n_{\mathrm{b}} = e^{O(\sqrt{n})}$. We prove the following lemma similarly to the proof of Lemma 8 in [L].

LEMMA 5.1. *Given a binary SSG* $(V, E = S \cup D \cup AA)$ *that halts with probability* 1, *let* (S, ω) *be its corresponding* d*-dimensional LP-type problem. Let* $n_{\mathrm{b}}(d)$ *denote the expected number of basis calculations required for Function lptype applied on* (S, ω), *when* d *elements* (*edges*) *of a basis need to be selected. Then*

$$n_{\mathrm{b}}(d) \leq n_{\mathrm{b}}(d-1) + \frac{1}{d}\sum_{i=1}^{d-1} n_{\mathrm{b}}(i) + 1,$$

for $d > 1$ *and*

$$n_{\mathrm{b}}(1) \leq 1.$$

PROOF. Let B_0 be an initial basis such that there exists a strategy σ^0 for player 1 with $B_0 = S_{\sigma^0}$ (i.e., $B_0 = \{(x_1, \sigma^0(1)), \dots, (x_d, \sigma^0(d))\}$). Let $\overline{\sigma^0}$ be the complementary strategy of σ^0 (i.e., $\forall i = 1, \dots, d$; $\overline{\sigma^0}(i) = z_i$ where (x_i, z_i) is the unique edge in $X_i \backslash \{(x_i, \sigma^0(i))\}$. Let

$$\omega^i = \max_{B \subseteq S,\ |B|=d} \{\omega(B) \mid (x_i, \sigma^0(i)) \in B\}.$$

Let $\{i_1, \dots, i_d\}$ be a permutation of $\{1, \dots, d\}$ such that $\omega^{i_1} \geq \cdots \geq \omega^{i_d}$. Now suppose that at step 2(a), Function lptype applied on $H = S$ chooses at random edge $h = (x_{i_r}, \overline{\sigma^0}(i_r))$, so $(x_{i_r}, \sigma^0(i_r))$ must be in every basis for $H \backslash \{h\}$. Then by solving a subproblem where $d - 1$ elements of a basis need to be selected, it will reach a basis B' satisfying $\omega(B') = \omega^{i_r}$. Then since every basis calculation increases the value of the objective function, it can no longer make at step 2(c) a basis computation to a basis B'', which has $(x_{i_j}, \sigma^0(i_j)) \in B''$, for any $j > r$. Hence every basis for H must contain the edges $(x_{i_j}, \overline{\sigma^0}(i_j))$ for every $j > r$, so these edges are now fixed until the algorithm terminates. By the same argument, after at most one basis computation, the algorithm chooses which one of the edges $(x_{i_r}, \sigma^0(i_r))$, $(x_{i_r}, \overline{\sigma^0}(i_r))$ is contained in a basis for H (we note that edge $(x_{i_r}, \sigma^0(i_r))$ is chosen if and only if no basis computation was done). Therefore, after one top-level iteration requiring an expected number of basis computations not exceeding $n_b(d - 1) + 1$, the size of the problem that remains to be solved (i.e., the number of edges remaining to be chosen for a basis for H) is $r - 1$. Then the recurrence follows from the fact that all possible choices of r are equally likely. □

THEOREM 5.2. *A binary SSG* $(V, S \cup D \cup AA)$ *that halts with probability* 1 *is solvable in strongly* $e^{O(\sqrt{\min\{d,m\}})} \times$ *poly*(n) *expected time.*

PROOF. We proceed as in the proof of Theorem 4.1, but instead of finding an optimal counterstrategy by solving an LP-type problem, we solve an LP problem. In order to achieve the claimed running time we need to show that when Function lptype is applied on instances (S, ω) of the binary SSG, n_b equals $e^{O(\sqrt{d})}$, and that t_b is strongly polynomial in n. Lemma 5.1 coupled with Lemma 2.7 implies that $n_b = e^{2\sqrt{d-1}}$, so $n_b = e^{O(\sqrt{d})}$ as needed. The constants in the LP program we solve in each basis computation are 0, 1 and $\frac{1}{2}$, so the algorithm of Khachiyan [Kh] solves the LP program in strongly polynomial time. □

Again, due to Lemma 2.2 we get:

COROLLARY 5.3. *The decision problem corresponding to binary SSGs is solvable in strongly* $e^{O(\sqrt{\min\{d,m\}})} \times$ *poly*(n) *expected time.*

6. Concluding Remarks.

It is possible (and maybe even more natural) to formulate an SSG as a discrete LP-type problem (see the definitions of discrete LP-type and dual LP-type problems in [Ha2]). For every $D' \subseteq D$ and $S' \subseteq S$ we say that $G(S', D') =$

$(V, S' \cup D' \cup AA)$ is a sub-game of G if $G(S', D')$ is a sub-game of G with respect to edges outgoing from both min and max vertices. Let $\Lambda = \mathbb{R} \cup \{\infty, -\infty\}$ be as before. Let $\sigma(S', D')$, $\tau(S', D')$ be an arbitrary pair of optimal strategies in a given sub-game $G(S', D')$ of G. We define μ: $2^S \times 2^D \to \Lambda$ in the following way:

$$\mu(S', D') = \begin{cases} -\infty & \text{if} \quad \exists i \in \{1, \ldots, d\} \quad \text{s.t.} \quad X_i \cap S' = \emptyset, \\ \infty & \text{otherwise if} \quad \exists i \in \{1, \ldots, m\} \\ & \quad \text{s.t.} \quad N_i \cap D' = \emptyset, \\ \sum_{z \in V} v_{\sigma(S',D'),\tau(S',D')}(z) & \text{otherwise.} \end{cases}$$

From this definition we get that for all $S' \subseteq S$, $\mu(S', D) = \omega(S')$ and (S, ω) is a d-dimensional LP-type problem. Moreover, for every $D' \subseteq D$, $\mu(S, D') = -\nu(D')$ and $(D, -\nu)$ is an m-dimensional *dual* LP-type problem. Hence, (S, D, μ) is a (d, m)-dimensional discrete LP-type problem. We solve this problem in the same time bound stated in Theorem 4.1 by applying the discrete LP-type algorithm in [Ha2]. It is interesting to apply the LP-type and discrete LP-type frameworks in order to achieve subexponential solutions to problems in fields other than game theory.

One major open problem is to develop polynomial time algorithms to solve the games studied in this paper. Another major open problem is to find a Nash equilibrium in Bimatrix Games in subexponential time [St]. This problem, together with factoring has been called “the most important concrete open question on the boundary of P today” [Pa]. We hope that the LP-type and discrete LP-type frameworks will shed new light on this problem.

Acknowledgments. I thank Uri Zwick who brought the paper by Ludwig [L] to my attention, and Leo Ruest for pointing out the reference [GW2].

References

[A] N. Amenta, Helly-type theorems and generalized linear programming, *Discrete & Computational Geometry* **12**, 241–261 (1994).

[BCJ+] A. Browne, E.M. Clarke, S. Jha, D.E. Long and W. Marrero, An improved algorithm for evaluation of fixpoint expressions, *Theoretical Computer Science* **178**(1–2), 237–255 (1997).

[BSV1] H. Björklund, S. Sandberg and S. Vorobyov, Randomized subexponential algorithms for parity games, Technical Report 2003-019, Department of Information Technology, Uppsala University (2003). Available online at http://www.it.uu.se/research/reports

[BSV2] H. Björklund, S. Sandberg and S. Vorobyov, A discrete subexponential algorithm for parity games, *Proceedings of the* 20*th Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science 2607, pp. 663–674, Springer-Verlag, Berlin (2003).

[BSV3] H. Björklund, S. Sandberg and S. Vorobyov, A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games, *Discrete Applied Mathematics* **155**, 210–229 (2007).

[Cl] K. L. Clarkson, Las Vegas algorithms for linear and integer programming when the dimension is small, *J. ACM* **42**(2), 488–499 (1995).

[CM] B. Chazelle and J. Matoušek, One linear-time deterministic algorithms for optimization problems in fixed dimension, *Journal of Algorithms* **21**, 579–597 (1996).

[Co] A. Condon, The complexity of stochastic games, *Information and Computation* **96**, 203–224 (1992).
[D] C. Derman, *Finite State Markovian Decision Processes*, Academic Press, New York (1972).
[EJS] E.A. Emerson, C.S. Jutla and A.P. Sistla, On model-checking for fragments of ν-calculus, in Costas Courcoubetis, editor, *Computer Aided Verification*, 5*th International Conference* (*CAV* '93), Lecture Notes in Computer Science 697, pp. 385–396, Springer-Verlag, Berlin.
[GW1] B. Gärtner and E. Welzl, Linear programming - randomization and abstract frameworks, *Proceedings of the* 13*th Annual Symposium on Theoretical Aspects of Computer Science*, pp. 669–687 (1996).
[GW2] E. Grädel and T. Wofgang, *Automata*, *Logics*, *and Infinite Games*, Lecture Notes in Computer Science 2500, Springer-Verlag, Berlin (2002).
[Ha1] N. Halman, An EGLP formulation for the simple stochastic game problem or a comment on the paper: "A subexponential randomized algorithm for the simple stochastic game problem" by W. Ludwig, Technical Report RP-SOR-01-02, Department of Statistics and Operations Research, School of Mathematical Sciences, Tel Aviv University (2001).
[Ha2] N. Halman, On the power of discrete and of lexicographic Helly-type theorems, *Proceedings of the* 45*th Annual Symposium on Foundations of Computer Science*, pp. 463–472 (2004).
[Ho] R. A. Howard, *Dynamic Programming and Markov Processes*, M.I.T. Press, Cambridge, MA (1960).
[J1] M. Jurdzinski, Deciding the winner in parity games is in UP∩co-UP, *Information Processing Letters* **68**, 119–124 (2000).
[J2] M. Jurdzinski, Small progress measures for solving parity games, Lecture Notes in Computer Science 1770, pp. 290–301, Springer-Verlag, Berlin (2000).
[Ka] G. Kalai, A subexponential randomized simplex algorithm, *Proceedings of the* 24*th Annual ACM Symposium on the Theory of Computation*, pp. 475–482 (1992).
[Kh] L.G. Khachiyan, A polynomial algorithm in linear programming, *Soviet Mathematics Doklady* **20**, 191–194 (1979).
[L] W. Ludwig, A subexponential randomized algorithm for the Simple Stochastic Game problem, *Information and Computation* **117**, 151–155 (1995).
[MC] M. Melekopoglou and A. Condon, On the complexity of the policy iteration algorithm for stochastic games, Technical Report TR 941, University of Wisconsin (June 1990).
[MSW] J. Matoušek, M. Sharir and E. Welzl, A subexponential bound for linear programming, *Algorithmica* **16**, 498–516 (1996).
[Pa] C.H. Papadimitriou, Algorithms, games, and the internet, *Proceedings of the* 33*rd Annual ACM Symposium on the Theory of Computation*, pp. 749–753 (2001).
[Pu] A. Puri, Theory of hybrid systems and discrete event systems, Ph.D. dissertation, EECS, University of Berkeley (1995).
[PV] H.J.M. Peters and O.J. Vrieze, *Surveys in Game Theory and Related Topics*, CWI Tract 39, Centrum voor Wiskunde en Informatica, Amsterdam (1987).
[Se] H. Seidel, Fast and simple nested fixpoints, *Information Processing Letters* **59**, 303–308 (1996).
[Sh] L.S. Shapley, Stochastic games, *Proceedings of the National Academy of Sciences*, *U.S.A*. pp. 1095–1100 (1953).
[St] B. Von Stengel, Exponentially many steps for finding a Nash equilibrium in a bimatrix game, *Proceedings of the* 45*th Annual Symposium on Foundations of Computer Science*, pp. 258–267 (2004).
[SW] M. Sharir and E. Welzl, A combinatorial bound for linear programming and related problems, *Proceedings of the* 9*th Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science 577, pp. 569–579, Springer-Verlag, Berlin (1992).
[ZP] U. Zwick and M. Paterson, The complexity of mean payoff games on graphs, *Theoretical Computer Science* **158**, 343–359 (1996).