DIMACS Technical Report 2005-13
April 1, 2005

Controlled Linear Programming for Infinite Games

by

Henrik Björklund, Ola Svensson, Sergei Vorobyov [1] [,2] [,3] [,4]

[1]DIMACS visitors. DIMACS hosts: Leonid Khachiyan and Endre Boros.
[2]Uppsala University, Information Technology Department, Box 337, 751 05 Uppsala, Sweden
[3]Supported by the Grant IG2003-2 067 from the Swedish Foundation for International Cooperation in Research and Higher Education (STINT, `www.stint.se`) and the Swedish Research Council grants "*Infinite Games: Algorithms and Complexity*", "*Interior-Point Methods for Infinite Games*", and "*Infinite Games for Program Verification: Combinatorial Optimization Approach*"
[4]Email: `vorobyov@csd.uu.se`

DIMACS is a collaborative project of Rutgers University, Princeton University, AT&T Labs–Research, Bell Labs, NEC Laboratories America and Telcordia Technologies, as well as affiliate members Avaya Labs, HP Labs, IBM Research, Microsoft Research and Stevens Institute of Technology. DIMACS was founded as an NSF Science and Technology Center.

ABSTRACT

We investigate the Controlled Linear Programming Problem (CLPP), a new combinatorial optimization problem nicely merging linear programming with games. In a system of linear constraints of the form $x_i \le p_i^j(\bar{x}) + w_i^j$, where p_i^j are linear homogeneous polynomials with nonnegative coefficients and $w_i^j \in \mathbb{R}$, some variables are distinguished as *controlled* and we want to select *exactly one* constraint for each controlled variable to make $\max \sum x_i$, subject to the remaining constraints, as large as possible (over all possible such selections). We suggest a number of iterative strategy improvement policies, simplex-like and interior-point. For many important cases we prove optimality conditions (when a local optimum is global), describe a number of combinatorial randomized *subexponential* algorithms, and show that the decision version of the problem is in NP ∩ coNP.

It turns out that the well-known mean payoff, discounted payoff, simple stochastic, and parity games, as well as the recent Longest-Shortest Paths (LSP) problem [10] (all in NP ∩ coNP, unknown to be in P) are easily reducible to the CLPP. This suggests a simplifying and unifying view of all these games as particular restricted instances of the CLPP, one of the "most general" problems in NP ∩ coNP to which all the known games in the class reduce (in certain sense "complete" in the class). We show that a slight generalization of the CLPP allowing for negative coefficients in the constraint polynomials p_i^j is NP-hard. So are the controlled versions of many other polynomial optimization problems, like Maximum Flow.

The simple algebraic and combinatorial structure of the CLPP unifies and gives insights into algorithmic and complexity-theoretic properties of infinite games, and is amenable to the powerful tools from combinatorial and polyhedral optimization, as well as linear programming. In this paper we investigate boundedness, CLPP duality, stability, optimality conditions, correctness of subexponential algorithms, and conditions for NP ∩ coNP-membership. In particular, strong duality implies NP ∩ coNP-membership, polynomial optimality conditions, and strongly subexponential algorithms.

Contents

1 Introduction **3**

2 The Controlled Linear Programming Problem **6**
2.1 Primal CLPP . . . 6
2.2 Longest-Shortest Paths as a CLPP . . . 7
2.3 Games as CLPPs . . . 7
2.3.1 Mean Payoff Games (MPGs) . . . 7
2.3.2 Discounted Payoff Games (DPGs) . . . 8
2.3.3 Simple Stochastic Games (SSGs) . . . 9
2.4 Dual CLPP . . . 10
2.5 Terminology and Notation . . . 10
2.6 Admissibility of Strategies . . . 11
2.7 Monotonicity of Iterative Improvement. Switches. Attractiveness and Profitability . 11
2.8 Attractiveness and Profitability: Unbounded Target Value . . . 12
2.9 Stability and Optimality, Subexponential Algorithms . . . 13
2.10 NP-Hardness for Non-Monotone CLPPs . . . 13
2.11 Controlled Max Bipartite Matching and Max Flow . . . 14

3 Tightness of Optimal Solutions and Existence of Optimal Counterstrategies **15**
3.1 Tightness of Finite Optimal Solutions . . . 15
3.2 Existence of an Optimal Counterstrategy . . . 15

4 When Stability Implies Optimality **17**
4.1 Strong Boundedness, Uniqueness, Duality . . . 17
4.1.1 Strong Boundedness . . . 17
4.1.2 Minimax Principle . . . 18
4.1.3 Stability and Optimality for Strongly Bounded Systems . . . 18
4.1.4 Anti-Attractive Switches Needed in Unbounded Systems . . . 20
4.1.5 Weak Duality . . . 20
4.1.6 Strong Boundedness is Essential for Weak Duality . . . 21
4.1.7 Strong Duality . . . 21
4.2 Stability Implies Optimality for Discounted Systems . . . 21
4.3 Stability Implies Optimality for the Univariate, Unit Coefficient Case . . . 22
4.3.1 Zero-Loop-Free Condition is Essential . . . 23
4.4 LSP-Duality . . . 24
4.4.1 LSP on DAGs . . . 24
4.4.2 LSP on Graphs Obtained from Ergodic MPGs . . . 24

5 Algorithms **26**
5.1 Discrete Switching Algorithms . . . 26
5.1.1 Single and Multiple Switch Algorithms . . . 26
5.1.2 Randomized Subexponential Algorithms . . . 26
5.2 Interior Point Methods . . . 28
5.2.1 Attractiveness and Profitability for Interior Point Switches, Discretization . . 29

6 Optimality Conditions **31**
6.1 Nonsingularity: A Necessary Condition of Boundedness 31
6.2 Bounded Systems Have Unique Solutions 32
6.3 Nonsingularity Does not Imply Boundedness 32
6.4 Sufficient Condition for Boundedness 32
6.5 Rational Systems 33

7 Membership in NP ∩ coNP **35**
7.1 Systems with Integer Coefficients 35
7.2 Rational Univariate Systems 38
7.2.1 Notes on Precision 40

8 Conclusions **41**

1 Introduction

Algorithmic and complexity-theoretic studies of infinite games can benefit a great deal from combinatorial optimization. In [7, 8, 10, 3, 2] we systematically illustrated this viewpoint by inventing the first randomized subexponential algorithms for parity, mean payoff, and simple stochastic games of arbitrary outdegree. Linear programming and polyhedral combinatorics turn out to be useful tools in exploring game structure and developing efficient combinatorial algorithms for games. In this paper we investigate a new *Controlled Linear Programming Problem* (CLPP) [3, 2], nicely merging linear programming with games, capturing all the games mentioned above, and allowing for their uniform algorithmic treatment.

For a general definition of a *controlled optimization problem*, consider a succinctly represented family $\mathcal{F} = \{G_i\}_{i \in I}$ of instances of a polynomially solvable optimization problem. (Each G_i is obtained from a generic instance G by fixing some controls.) Find an instance with the *largest* possible value of the associated objective function. We are better off with a concrete example.

Longest Shortest Paths. Let G_i's be weighted digraphs with source s and sink t. Find a graph with the *longest* shortest s-t-distance (counting negative weight loops as $-\infty$). As a succinct representation, take a digraph G with a distinguished subset of *controlled* vertices C, and generate G_i's by selecting exactly one outgoing edge per controlled vertex. This gives a compact representation of an exponential in $|C|$ family of graphs. This problem, which we called the *Longest Shortest Paths (LSP)* problem, has useful applications, e.g., for solving mean payoff games and is first investigated in [10].

The example of a *controlled* problem above, in which the controller selects options in order to optimize the target function over all possible selections, seems deceptively simple to generalize. Take Max-Bipartite Matching or Max-Flow and allow the controller to choose edges from some vertices so as to minimize the resulting maximum matching/flow. These natural controlled problems turn out to be NP-hard (Section 2.10). Surprisingly, the LSP and CLPP are fundamentally different: under reasonable assumptions both belong to $\mathrm{NP} \cap \mathrm{coNP}$ and we suggest *subexponential* algorithms.

The following questions arise immediately. Can we equip the family $\mathcal{F}$ with a meaningful *polynomial neighborhood structure* and corresponding target/quality function[1] allowing for a *polynomial local search* (PLS)? Combinatorially, we get a directed graph with family members $G_i \in \mathcal{F}$ as vertices, and edges corresponding to quality improvements. How fast can we find local maxima in this graph? It turns out that our chances are better than exponential (i.e., *subexponential*) in the LSP problem [10]. How about the quality of local maxima in this structure? How can we verify that a given local maximum is global? Can we guarantee that *every* local maximum is global? All those questions have satisfactory positive answers in the case of the LSP problem, which resulted in the first combinatorial subexponential algorithm for mean payoff games [10].

Now the main question. Are there any other useful *controlled* optimization problems with

[1] Note that there are target functions in each instance, but here we are talking about the "second-order" target function comparing the quality of *different* instances.

positive (satisfactory) answers to the questions above? Do they have good and interesting applications? How far can we go in generalizing?

In this paper we give a positive answer by suggesting and investigating another (new) "universal" problem of this kind, which we dubbed the *Controlled Linear Programming Problem* (CLPP) (see the definition above; first introduced in [3, 2]). It turns out that the well-known mean payoff, discounted payoff, and simple stochastic games [16, 18, 13, 29], as well as the LSP problem and parity games [10], are easily reducible to the CLPP. This suggests a simplifying and unifying view of all these games as particular restricted instances of the problem, and proves its usefulness ("completeness"). Our combinatorial randomized subexponential algorithms for the CLPP (Section 5) are immediately applicable to solve all the above games.

The CLPP demonstrates a robust structure amenable to a polynomial local search (PLS). Looking at unselected controller constraints, which have less restrictive right-hand sides in the optimal solution, compared to the current selection, allows for defining a numerous monotonic iterative improvement schemes (discrete and *interior point* attractive switches; Sections 2.7, 5.2). In many interesting cases of the CLPP, we also prove that a stable (not locally improvable) pure strategy (selection of constraints) is also globally optimal (Section 4). As a result, the problem of finding global maxima for the CLPP in these cases can be solved in *subexponential* time (Section 5), and the corresponding decision version (is there a selection providing for the optimum value above the threshold) is in $\mathrm{NP} \cap \mathrm{coNP}$ (Section 4). We also investigate the CLPP with nonnegative integer coefficients and the rational univariate CLPP, both in $\mathrm{NP} \cap \mathrm{coNP}$, but without the "stability $\Rightarrow$ optimality" property and without currently known subexponential iterative improvement algorithms (Section 7). This shows, in particular, that the CLPP is a far *more general* problem in $\mathrm{NP} \cap \mathrm{coNP}$ than any of the games mentioned above, and suggests that it is "complete" for a certain class of games.

The next natural step in generalization leads to (presumable) intractability. We show that a slight generalization of the CLPP, allowing for negative coefficients in the polynomials p_i^j, leads to NP-hardness. As a result, we have a sufficiently sharp classification of $\mathrm{NP} \cap \mathrm{coNP}$ and NP cases in terms of nonnegative/arbitrary coefficients in the right-hand sides of the CLPP constraints. Our examples of the Controlled Bi-Partite Matching and Controlled Max-Flow confirm this classification (Section 2.11).

The CLPP is more general than the games discussed above, and it is not known whether it can be optimized, in general, in expected subexponential time or even belongs to $\mathrm{NP} \cap \mathrm{coNP}$. In [3] and here we show that this is the case for a number of possible extensions of the games. We investigate conditions on the structure of the constraints sufficient for subexponential solvability and membership in $\mathrm{NP} \cap \mathrm{coNP}$. To this end, we also define a dual minimization version of the CLPP, and study how the relation between an instance and its dual can be used to infer boundedness of an optimal solution.

The crucial property underlying the subexponential iterative optimization for CLPP is "stability $\Rightarrow$ optimality". A strategy (choice of constraints), which cannot be improved by local changes, is called *stable* (stability is polynomial time verifiable). Subclasses of

the CLPP for which stable strategies are guaranteed to be optimal belong to NP ∩ coNP, and can be solved in expected subexponential time. A number of CLPP classes satisfying "stable ⇒ optimal", covering all games mentioned above, are investigated in [3]. For other subclasses, the existence of polynomial optimality conditions is closely linked to verifying the boundedness of a system, studied in [3] and here. In this paper, we derive the desired "stability ⇒ optimality" property using *duality* and *boundedness* considerations. This leads to conceptually easier and more transparent proofs, compared to [3], based on potential transformations of hypergraphs and convergence of infinite series. Certainly, introducing the Dual CLPP and investigating it together with the Primal makes the theory more symmetric and leads to a better understanding of the problem. As a result, we close certain gaps in the previous development [3], like correctness of attractive switches, and existence of optimal counterstrategies. We also find simple yet fundamental relations between linear algebra concepts and games, including nonsingularity, Farkas lemma, etc.

Linear Complementarity Problem (LCP). The LCP consists in finding vectors $w, z \geq 0$ satisfying $w = Mz + q$ and $w^T \cdot z = 0$ [21, 15]. In [12] we describe new, LCP-based, algorithms (pseudopolynomial and subexponential) for mean payoff games, which can be extended to cover some cases of the CLPP as well. It should be noticed that LCP instances arising from infinite games and CLPP instances are *not* P-matrix LCPs (all principal minors positive). Therefore, the CLPP is *not reducible* to the P-matrix LCP, which is another member of NP ∩ coNP (either show that the matrix is non-P, or find a unique solution to the LCP). Currently there are no known subexponential algorithms for P-matrix LCPs, neither for other classes of non-polynomial LCPs (except [12]), so the existing LCP-theory is not very helpful. In contrast, we describe such algorithms for the CLPP in this paper.

Novelty. The results presented are new. After discovering first subexponential algorithms for parity [7], mean payoff games and the LSP problem [10], simple stochastic games of *arbitrary* outdegree [8, 4, 11] (see Section 5.1), and *completely unimodal* pseudoboolean optimization [5, 6], we were looking for the unifying superproblem to which all the above would easily reduce. It is pleasant to see that the CLPP, marrying games with linear programing in a simple and natural way, is exactly this common unifier. It simplifies the combinatorial structure underlying infinite games and adds yet another tantalizing problem to the mysterious and sparse class NP NP ∩ coNP of *well-characterized* (J. Edmonds) problems unknown to be in P.

2 The Controlled Linear Programming Problem

In this section we present the controlled linear programming problem (CLPP) and define its dual minimization counterpart. The problem was first introduced and studied in [3, 2]. We also review some fundamental properties of the CLPP.

2.1 Primal CLPP

Let $x = (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m})$ be a vector of variables. The first n variables are called *controlled*, and the remaining m *uncontrolled*. By p_i^j we denote linear *homogeneous* polynomials with *nonnegative* coefficients and variables in x, and by w_i^j real numbers. A *monotone* linear constraint has form $x_i \leq p_i^j + w_i^j$. Such a constraint is called:

1. *optional*, if x_i is a controlled variable, and
2. *compulsory*, if x_i is an uncontrolled variable.

Let S be a system of monotone constraints with at least one constraint per variable x_i in x. A *controller strategy* σ consists in selecting *exactly one* constraint for each controlled variable.[2] Denote by $S(\sigma)$ the resulting system of constraints obtained by deleting from S all optional constraints not selected by σ.

Definition 2.1 *An instance of the* Controlled Linear Programming Problem *(CLPP for short) is specified as follows.*

Given: *a system of monotone constraints S.*

Find: *a controller strategy σ^** maximizing $\max_x(1^T x | S(\sigma^*))$, *over all such strategies.*

Remark. We can make it more general by allowing the target function of the form $c^T x$ for $c^T \geq 0$. However, to provide for a strict monotonic improvement (Section 2.7) we need to actually optimize the function $c^T x + 1^T x$. □

Let S_i be the set of optional constraints for controlled variable x_i. The space of (pure positional) strategies for the controller is isomorphic to the Cartesian product

$$\mathcal{P} = \prod_{i=1}^{n} S_i.$$

To each strategy σ in $\mathcal{P}$ we assign the value $\max(1^T x | S(\sigma))$, thus obtaining a function from $\mathcal{P}$ to $\mathbb{R} \cup \{-\infty, \infty\}$. Solving the CLPP amounts to finding a maximum of this function. Note that this function is generally *nonconvex*.

[2] Such strategies are called *pure*, in contrast to *mixed*, i.e., do not use randomness. Section 5.2 deals with *mixed* strategies for the CLPP. In Section 2.4 we define an environment strategy symmetrically, as a selection of exactly one constraint for each uncontrolled variable. In the sequel the controller and the environment will more often be referred to as players MAX and MIN, respectively.

2.2 Longest-Shortest Paths as a CLPP

The *Longest-Shortest Paths* (LSP) problem is a controlled optimization problem introduced in [10] as a tool for solving mean payoff games. A problem instance is a weighted digraph G with a distinguished sink t and a subset C of controlled vertices. The controller (MAX) selects exactly one outgoing edge from each controlled vertex, with the goal of making the shortest distances from all vertices to t as long as possible. Given an instance and a strategy σ of MAX in the controlled vertices, we assign values to vertices in the following way. Let G_σ be the graph obtained by removing all edges leaving controlled vertices, except those selected by σ. If a negative weight cycle is reachable from v in G_σ, then v gets value $-\infty$. If only positive value loops are reachable from v, and t is not, then v gets value $+\infty$. Otherwise, we assign v the value of the shortest path from v to t in G_σ.[3]

In our reduction of mean payoff games to the LSP problem [10], MAX can secure a value > 0 in the game exactly from the vertices for which he can enforce the $+\infty$ path (loop) in the LSP. Finding ergodic partitions and exact values reduce to this problem [10]. More on this reduction and how the symmetry of mean payoff games helps to solve them using a minimization version of LSP can be found in Section 4.4.

Every LSP instance G can be rewritten as a CLPP instance S. For every edge (u, v) with weight w in G, let $u \leq v + w$ be a constraint in S. If $u \in C$, the variable u is controlled, otherwise it is uncontrolled. Finally, add the compulsory constraint $t \leq 0$. The value of a variable u in an optimal solution for S equals the longest shortest path MAX can achieve for u in G. Thus the LSP is a simple particular case of the CLPP.

In practical algorithms for mean payoff games and LSP instances, it is undesirable to solve a linear program in each iteration, since more efficient graph algorithms are applicable. The CLPP setting is, however, useful for reasoning about the problems since it makes many proofs considerably easier; see, e.g., Section 2.7.

2.3 Games as CLPPs

Let us further illustrate the definition and usefulness of the CLPP by representing some well-known games.

2.3.1 Mean Payoff Games (MPGs)

Mean Payoff Games (MPGs) [16, 18, 29, 26, 10] are full-information infinite adversary games on finite directed leafless[4] edge-weighted graphs $G = (V, E, w)$, with $V = V_{\max} \cup V_{\min}$ and $w : E \to \{-W, \ldots, W\} \subset \mathbb{Z}$. Starting in a vertex v_0 players Max and Min select edges (if $v_i \in V_{\max}$ then Max selects an edge from v_i and moves to its destination vertex v_{i+1}, otherwise Min chooses) and as a result construct an infinite sequence of edges (v_i, v_{i+1}).

[3] Zero-weight loops constitute a special case, which complicates matters. Such loops can be avoided; see below and [10] for details.

[4] containing no sinks, i.e., every vertex has an outgoing edge.

Player Max/Min wants to maximize/minimize, respectively,

$$\liminf_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k-1} w(v_i, v_{i+1}), \quad \text{and} \quad \limsup_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k-1} w(v_i, v_{i+1}).$$

It turns out that MPGs are solvable in pure positional strategies for both players, every vertex has a value, which both players can secure by applying these strategies, and consequently the problem whether this value is above/below a certain threshold is in NP ∩ coNP [16, 18]. It is convenient and sufficient to solve the *0-mean partition problem*: find a vertex subset $V^+ \subseteq V$ from which Max can secure a positive value. Other MPG problems, including finding exact values, ergodic partitioning, optimal strategies, are reducible to 0-mean partitioning, using dichotomy, edge re-weighting, and approximation [18, 10].

In [10] we solve the 0-mean partition by reduction to the LSP problem. Add a *sink* vertex to the game graph and *retreat* edges of weight 0 from every MAX vertex. Now player MAX can always secure, if he wants, distance 0 to the sink from each of his vertices. By monotonically iteratively improving his initial "retreat to the sink everywhere" strategy, we can figure out whether Max's best strategy secures a finite, or a $+\infty$ distance to the sink. The latter corresponds exactly to the Max's ability to enforce a *positive* loop with vertices in V^+. Thus 0-mean partition for MPGs is easily representable (through LSP) as a CLPP. There is a subtle catch here. In an MPG, enforcing a 0-weight loop does not give MAX vertices in V^+. In the CLPP a 0-loop like $\{x \le y, y \le x\}$ gives target value $\max(x + y) = +\infty$. To avoid this discrepancy, before solving 0-mean partition via CLPP we first transform the game into a form *without* 0-weight loops. Multiply all edge weights by $n + 1$ (n number of vertices) and subtract 1. All 0-weight loops will become negative (no good for MAX) and all other will preserve their signs. See also Section 4.3.

2.3.2 Discounted Payoff Games (DPGs)

Let $\delta \in (0, 1)$ be given. Consider the game $G = (V = V_{\text{MIN}} \cup V_{\text{MAX}}, E, w)$ on a leafless edge-weighted graph. The objective of the adversaries MAX and MIN selecting outgoing edges (moves) from V_{MAX} and V_{MIN} is to maximize/minimize (in the limit) the value (see [29] for details)

$$(1-\delta)\sum_{i=0}^{i=\infty} \delta^i \cdot w(e_i). \tag{1}$$

It has been shown that the optimal value vector of a DPG is the unique solution to the set of equations [29]:

$$x_i = \begin{cases} \max_{(i,j)\in E}\{(1-\delta)w_{ij} + \delta x_j\} & \text{if } \ i \in V_{\text{MAX}}, \\ \min_{(i,j)\in E}\{(1-\delta)w_{ij} + \delta x_j\} & \text{if } \ i \in V_{\text{MIN}}. \end{cases}$$

It is straightforward to see that this corresponds to the following CLPP:

Maximize $\sum_i v_i$ subject to

compulsory constraints:
$v_i \le (1-\delta)(w_{ij} + \delta v_j)$, *for* $v_i \in V_{\text{MIN}}$ *and* $(v_i, v_j) \in E$,

optional constraints:
$v_i \le (1-\delta)(w_{ij} + \delta v_j)$, *for* $v_i \in V_{\text{MAX}}$ *and* $(v_i, v_j) \in E$.

Actually, the normalizing factor $(1-\delta)$ in (1) is introduced just for technical convenience. We may drop it and generalize by allowing a different discounting factor $\delta_i \in (0,1)$ for each edge of the game graph. Such a generalized DPG will be easily represented as a CLPP. Note the special form of the DPG-generated CLPP: it contains one variable per constraint side, discounted on the right.

2.3.3 Simple Stochastic Games (SSGs)

We now show how CLPPs express *simple stochastic games* (SSG), a subclass of stochastic games introduced and studied by Shapley [28] (see [13, 14, 17] for a thorough treatment). A simple stochastic game is given by a directed graph $G = (V \cup \{0,1\}, E)$, where the distinguished vertices 0, 1 are called the 0-*sink*, the 1-*sink*, and the vertex set V is partitioned into subsets V_{MAX}, V_{MIN}, V_{avg} of the maximizing player MAX, minimizing player MIN, and probabilistic ones. Starting in the initial vertex players move a pebble selecting outgoing edges from their vertices. When the pebble comes to a probabilistic vertex, one of the outgoing edges is randomly selected, according to a specified probability distribution. The play stops when the pebble comes to a sink. Player MAX wants to maximize and player MIN to minimize the probability of getting to the 1-sink. SSGs turn out to possess a *value* ν and are solvable in pure positional strategies, optimal for both players, securing them the equilibrium probability ν of getting to the 1-sink.

A natural decision problem consists in determining, given an SSG and a probability $p \in [0,1]$, whether the value of the game is $\ge p$. Condon [13] showed that this problem is in NP NP$\cap$coNP, and suggested several algorithms for finding values and optimal strategies. Actually, the first step Condon [13] applies is polynomial reduction (corrected in [29]) of an arbitrary SSG to a *stopping* SSG with a negligibly different value. Say that an SSG *stops with probability* 1 if no matter how players select moves, the game comes to a sink with probability 1. Roughly (see [13, 29] for details), every edge (u, v) with assigned probability p in the original game[5] transforms into two edges: (a) one going to the 0-sink, with probability pq, and (b) the other, with probability $p(1-q)$, from u to v, where q is an appropriately selected exponentially small probability.

Suppose an SSG has undergone such a transformation. After throwing away all edges of type (a) the corresponding instance of the controlled LP problem is easy to construct. Write

[5]Edges of players MAX and MIN are initially given probability 1.

optional constraints $x \leq p \cdot y$ for every edge (x, y) from V_{MAX}, and compulsory constraints $y \leq p \cdot x$ for every edge from V_{MIN}, where $p < 1$ is the probability assigned to the edge in the transformation above. If (u, v_i) are edges from a vertex in V_{avg} with assigned probabilities p_i, $i \in I$, then write a compulsory constraint $u \leq \sum_{i \in I} p_i \cdot v_i$ (note that $\sum_{i \in I} p_i < 1$ by construction). Add compulsory constraints $v_0 \leq 0$ and $v_1 \leq 1$ for sinks.

Clearly, the resulting controlled LP is just a particular case of the discounted multivariate case considered in Section 4.2.[6] An additional advantage is that we do not need to transform the game into a binary one, which may increase the number of vertices quadratically, and turn a subexponential (in the number of variables) algorithm into an exponential one; see Section 5.1. Since every polynomial defining a constraint, except for the 1-sink, is homogeneous, any choice of optional constraints will yield a feasible linear program. In fact, for any such LP, setting all variables to zero gives a feasible solution, so there is no problem with selecting an initial admissible strategy; see Section 2.6.

2.4 Dual CLPP

Duality is a powerful and useful concept in combinatorial optimization. The CLPP makes no exception, and we develop the basic duality theory in this paper. It allows us, in particular, to easily derive optimality conditions for stable strategies in Section 4.

Call a linear constraint of the form $x_i \geq p_i^j + w_i^j$ *anti-monotone*. As before, such a constraint is *optional* if x_i is controlled and *compulsory* otherwise. Let S^d be a system of anti-monotone constraints with at least one constraint per variable x_i in x.

Symmetrically, an *environment strategy* τ is a selection of exactly one constraint for each *uncontrolled* variable. Denote by $S^d(\tau)$ the resulting system of constraints obtained by deleting from S^d all compulsory constraints not selected by τ.

Definition 2.2 *An instance of the* Dual Controlled Linear Programming Problem *(DCLPP for short) is as follows.*

Given: *a system of anti-monotone constraints* S^d.

Find: *an environment strategy* τ^* minimizing $\min_x(1^T x | S^d(\tau^*))$, *over all environment strategies.* □

Remark. Note that the concept of CLPP duality differs drastically from the one accepted in Linear Programming (see, e.g., [27]), but as we will see, the duality theory will be congenial. □

2.5 Terminology and Notation

In the sequel the controller and the environment are referred to as players MAX and MIN, respectively. Throughout the paper we systematically use Greek letters σ, τ for MAX's and

[6] Actually, Shapley in the original definition of SGs stipulated discountedness [19].

MIN's strategies. Given a constraint system S we denote its dual by S^d. If both MAX and MIN fix their strategies σ and τ then the resulting systems are denoted by $S(\sigma,\tau)$ and $S^d(\sigma,\tau)$. For a system $S(\sigma,\tau)$ or $S^d(\sigma,\tau)$ denote by $S^=(\sigma,\tau)$ the system obtained by replacing all inequalities by *equalities*. The optimal values of the linear program associated with a constraint system are denoted as follows:

$$\begin{array}{rcl@{\qquad\qquad}rcl} opt(S(\sigma,\tau)) & =_{df} & \max(1^T x|S(\sigma,\tau)), & opt(S(\sigma)) & =_{df} & \max(1^T x|S(\sigma)), \\ opt(S^d(\sigma,\tau)) & =_{df} & \min(1^T x|S^d(\sigma,\tau)), & opt(S^d(\tau)) & =_{df} & \min(1^T x|S^d(\tau)). \end{array}$$

2.6 Admissibility of Strategies

A strategy σ of MAX is called *admissible* if $S(\sigma)$ is feasible. (It follows that for each strategy τ of MIN the system $S(\sigma,\tau)$ is feasible). Symmetrically, a strategy τ of MIN is admissible if $S^d(\tau)$ is feasible. Admissibility is a crucial property: all our algorithms iterate through and maintain admissibility of strategies at every step [10, 3, 2]; see Section 5.

2.7 Monotonicity of Iterative Improvement. Switches. Attractiveness and Profitability

Attractiveness (polynomially verifiable property) of strategy switches is a central concept in developing efficient iterative improvement local search algorithms for the CLPP; see Section 5. It allows us to define an appropriate polynomial neighborhood structure on the set of strategies and a multitude of iterative strategy improvement algorithms for the CLPP [3]. We first concentrate on the case of a finite target value and then in Section 2.8 make modifications to cover the unbounded case.

Definition 2.3 (Switch, Attractiveness, Stability) *For a strategy σ, a* single switch *changes the selected constraint for exactly one variable. Suppose the current strategy is σ, the value* $\max(1^T x|S(\sigma))$ *is finite, and variable assignment* $x^* = \arg\max(1^T x|S(\sigma))$. *Let $\sigma_i \equiv x_i \le p_i^\sigma(x) + w_i^\sigma$ be the constraint selected for the controlled variable x_i in σ, and let $\sigma_i' \equiv x_i \le p_i^{\sigma'}(x) + w_i^{\sigma'}$ be another constraint in S. The single switch from σ_i to σ_i' is called* attractive *if $p_i^\sigma(x^*) + w_i^\sigma < p_i^{\sigma'}(x^*) + w_i^{\sigma'}$. A strategy without attractive switches is called* stable. *Attractive switches and stable strategies for* MIN *in the DCLPP are defined symmetrically.* □

Remark. This definition tacitly assumes that x^* is the only optimal solution to $S(\sigma)$. Otherwise, attractiveness of a switch depends on a selected optimal solution. In fact, by Corollary 6.3, the system has only one solution. □

The concept of attractive switches is important, since they are also *profitable*:

Theorem 2.4 *Any combination of attractive switches for* MAX *in the CLPP instance S (or for* MIN *in the DCLPP instance S^d) is profitable, i.e., leads to a monotonic increase (decrease) of the target function $1^T x$.*

Proof. Assume that a number of attractive switches are made, from strategy σ to σ'. Let the variable assignment $x^* = \arg\max(1^T x | S(\sigma))$. For every switched variable x_i, we have the old constraint $x_i \leq p_i^\sigma(x) + w_i^\sigma$, and the new constraint $x_i \leq p_i^{\sigma'}(x) + w_i^{\sigma'}$. Since the switches are attractive and constraints are monotonic, it is easy to see that x^* is a solution to the new LP $S(\sigma')$, because every constraint is either the same or more generous, i.e., has a larger right-hand-side.

Take one switched variable x_i, let $x'_i = x^*_i + \varepsilon$, for a small enough $\varepsilon > 0$, and let $x'_j = x^*_j$ for all $j \neq i$. Then x' is a solution to the new LP $S^d(\sigma')$, because:

1. x_i does not violate its new constraint, which by attractiveness is more generous, and
2. the increase of x_i does not decrease the right-hand-side polynomials of any constraint, since all coefficients are nonnegative.

We have thus shown that the resulting strategy is admissible and that there exists a better solution to the new LP. Therefore, the switch is profitable. In addition, we see that the values of variables for which attractive switches are made are guaranteed to increase in an optimal solution to the new system. Note that the previous argument essentially uses the fact that the right-hand-side polynomials of constraints have nonnegative coefficients. It would fails if negative coefficients were allowed. The proof for MIN strategies in a dual system is symmetrical. □

Theorem 2.4 applies to any games with strategy evaluation functions expressible by CLPPs. Many earlier such proofs were considerably more complicated [7, 10].

2.8 Attractiveness and Profitability: Unbounded Target Value

Attractive switches improve the target function value, as long as it is finite. However, performing attractive switches makes sense even when the target function is unbounded. In many circumstances we want to maximize the values of *all* variables, rather than just their sum. Imagine z is unbounded, the current constraint for x is $x \leq 5$, and there is an unselected constraint $x \leq z$. Then switching from $x \leq 5$ to $x \leq z$ makes x unbounded as well.

Extending Definition 2.3 and Theorem 2.4 to the unbounded case can be done as follows. It suffices to define $\arg\max(1^T x | S(\sigma))$ as the vector of values for variables x containing the value $\arg\max(x_i | S(\sigma))$ whenever $\max(x_i | S(\sigma))$ is finite, and the value M big enough for variables x_j with unbounded $\max(x_j | S(\sigma))$. A big enough M can be selected as any number exceeding the largest of the finitely bounded right-hand sides in $S(\sigma)$.

It is also convenient to get rid of unbounded variables and associated constraints to simplify the system. Some care should be taken to avoid premature simplifications, and allow for making attractive switches for all eventually unbounded variables. Here is the recipe.

Theorem 2.5 *Eliminating unbounded variables and constraints from the system $S(\sigma)$ does not affect the optimal values of the remaining variables, provided that σ is stable.*

Proof. Assume the value of a controlled variable x_i is bounded in $S(\sigma)$. Since x_i has no attractive switches, every unbounded variable must have the coefficient 0 in every constraint for x_i. Therefore, x_i will not be affected by the elimination of the unbounded variables. Suppose the value of an uncontrolled variable y_i is bounded in $S(\sigma)$. Then it has a bounded constraint. Since the unbounded constraints will never tighten the value of y_i, even if the bounded constraints go unbounded, we can safely remove them. □

Eliminating unbounded variables allows us to assume w.l.o.g. in the premises of theorems below that a system has a finite optimal solution under a current strategy.

It is easy to construct examples where an elimination in a system with an unstable strategy may lead to a non-optimal solution. In the example above assume we eliminated z and $x \leq z$ before making a switch from $x \leq 5$.

2.9 Stability and Optimality, Subexponential Algorithms

Recall that a strategy is *stable* if it has no attractive switches. We have earlier showed that game problems with strategy evaluation functions such that *stable strategies are also optimal* can be solved in *randomized subexponential* time [8, 7, 10, 1]. It turns out that the same is true for the CLPP. Not all CLPP instances have this nice property, but a number of interesting subclasses do. This is studied in detail in Sections 4, 5.

Suppose we are given some combinatorial optimization problem, and want to know whether its decision version belongs to NP ∩ coNP and has subexponential algorithms. The CLPP and stability give us a useful tool. First, we try to write the problem as a CLPP, and then check if stable strategies are optimal. If they are, the problem whether the optimal value is $> B$ belongs to NP ∩ coNP. Indeed, a short witness for a YES-instance is a strategy providing for a value $> B$; a NO-instance is witnessed by a stable (hence optimal) strategy yielding a value $\leq B$.

As we discuss in Sections 4, 5, the subexponential analysis of the algorithms, adapted from combinatorial linear programming relies on the fact that once a local optimum of a subsystem (with a subset of the original constraints) has been found, we never need to return to a strategy in the subsystem. When stability does not imply optimality, this property no longer holds, and the algorithms might even fail to correctly find an optimum.

2.10 NP-Hardness for Non-Monotone CLPPs

In the definition of the CLPP we stipulate that all coefficients in the constraints are non-negative. This restriction is essential if we hope to solve instances efficiently. In fact, by reduction from 3-SAT we prove [3] the following

Theorem 2.6 *If negative unit coefficients are allowed in the right-hand sides of CLPP constraints then the problem becomes NP-hard.*

Proof. By reduction from 3-SAT. Given an instance ϕ of 3-SAT with the variables $(x'_1, ..., x'_n)$ and the clauses $(C'_1, ..., C'_m)$, construct a CLPP instance S with the variables $(x_1, ..., x_n)$, $(\bar{x}_1, ..., \bar{x}_n)$, and $(C_1, ..., C_m)$, where x_i represents x'_i, $\bar{x}_i$ represents $\neg x'_i$, and C_j represents the truth value of C'_j. For every variable x'_i in ϕ construct the compulsory constraints $x_i \leq 1$, $\bar{x}_i \leq 1 - x_i$. For every clause $C'_j = l_1 \vee l_2 \vee l_3$ in ϕ add the optional constraints $C_j \leq l_1$, $C_j \leq l_2$, $C_j \leq l_3$. The objective function to be maximized is $\sum_{i=1}^{n}(x_i + \bar{x}_i) + \sum_{j=1}^{m} C_j$. It is easy to see that every optimal solution will have $x_i + \bar{x}_i = 1$. Thus $\sum_{i=1}^{n}(x_i + \bar{x}_i) + \sum_{j=1}^{m} C_j = n + \sum_{j=1}^{m} C_j$.

Assume the optimal solution to S is $n+m$. Then $n + \sum_{j=1}^{m} C_j = n + m$ implies $\sum_{j=1}^{m} C_j = m$, and we know that $\forall i : (x_i \leq 1 \wedge \bar{x}_i \leq 1)$, which implies $\forall j : C_j \leq 1$. Then if $\sum_{j=1}^{m} C_j = m$, there must exist a variable in every clause that is equal to 1, i.e., ϕ is satisfiable.

Suppose ϕ is satisfiable. Then there exists a variable assignment satisfying every clause, i.e., $\forall j : C_j = 1$, which implies $\sum_{j=1}^{m} C_j = m$. Thus the optimal solution to the CLPP instance is $n + m$. □

2.11 Controlled Max Bipartite Matching and Max Flow

It might seem that any polynomial optimization problem can be easily turned into a controlled version. We show that usually this will result in an NP-hard problem.

In the Controlled Max Bipartite Matching problem, we are given a bipartite, undirected graph $G = (V = (A \cup B), E)$, a subset $C \subseteq A$ of controlled vertices, and a function $c : C \to \mathbb{N}$ that for each controlled vertex says how many incident edges the controller is allowed to remove, trying to *minimize* the resulting maximum matching. This problem is NP-complete. An optimal strategy of the controller can serve as a short witness. We show hardness by reduction from Set Cover. Given an instance (X, F, k) of Set Cover, where X is a set, $F \subseteq \mathcal{P}(X)$, and $k \in \mathbb{N}$, the question is whether there is a subset $S \subseteq F$ of size at most k, such that every element in X is a member of some set from S. To solve it we produce an instance of Controlled Max Bipartite Matching as follows.

The two vertex sets are X and F. An edge $(u, v) \in X \times F$ belongs to the edge set E iff $u \in v$. For each vertex $u \in X$ the controller can remove all but one adjacent edge $(c(u) = deg(u) - 1)$. Edges can be removed so that the maximum bipartite matching is $\leq k$ if and only if there was a set cover of size at most k in (X, F, k).

The more general, but still polynomial, Maximum Flow problem can be easily expressed by linear programs, but *not in the form of monotonic* constraint we allow in the CLPP. Skew symmetry and flow preservation constraints require negative coefficients (so Theorem 2.6 hints it may be NP-hard). Actually, no matter which representation tricks one tries to use, negative coefficients are unavoidable. Indeed, Controlled Maximum Flow is also NP-hard. In this problem, a flow network is given, together with a subset C of the vertices of the network. There is also a function $c : C \to \mathbb{N}$. The problem is to remove $c(v)$ or fewer outgoing edges from every $v \in C$ in a way that minimizes the maximum flow in the remaining network. NP-hardness can be proved by a reduction from the Controlled Max Bipartite Matching.

3 Tightness of Optimal Solutions and Existence of Optimal Counterstrategies

In this section we prove two basic results that we heavily rely upon in the sequel. The first one states that a finite optimal solution to a system $S(\sigma, \tau)$, where MAX and MIN have fixed their strategies, is *tight*, i.e., satisfies all constraints as equalities. The second one asserts that whenever MAX fixes a strategy, MIN can reply with an *optimal counterstrategy*.

3.1 Tightness of Finite Optimal Solutions

Lemma 3.1 *Let S be a CLPP instance and σ, τ be a pair of strategies such that $x^* = \arg\max(1^T x | S(\sigma, \tau))$ is finite. Then*

1. x^* is a feasible solution to $S^=(\sigma, \tau)$,

2. conversely, $1^T x' = 1^T x^*$ for every other feasible solution x' to $S^=(\sigma, \tau)$.

In other words, optimal solutions to $\max(1^T x | S(\sigma, \tau))$ are tight, and tight solutions are optimal.

Proof. If x^*, a feasible solution to $S(\sigma, \tau)$, did not solve the system $S^=(\sigma, \tau)$, it wouldn't be an optimal solution to $S(\sigma, \tau)$, since for some constraint, the inequality would be non-tight, and we could slightly increase the value of the corresponding variable, without violating any constraints (due to monotonicity). Suppose that $S^=(\sigma, \tau)$ has a solution x' such that $1^T x' \neq 1^T x^*$. Rewrite $S^=(\sigma, \tau)$ in matrix form as $Ax = b$. Since $Ax' = Ax^*$ we get $A(x' - x^*) = 0$. Let λ be any real number. Then $Ax^* + A\lambda(x' - x^*) = A(x^* + \lambda(x' - x^*)) = b$. Since $1^T x' \neq 1^T x^*$ this means that $Ax = b$ has arbitrarily large solutions, contradicting the boundedness assumption. □

3.2 Existence of an Optimal Counterstrategy

In parity, mean payoff, discounted payoff, and simple stochastic games, there is an *optimal counterstrategy* of the opponent against every strategy of the first player. Here we show that the same is true for the CLPP. This means that we can regard the problem not only as a combinatorial optimization problem, but as a game between two players, where one selects a strategy and the other replies with an optimal counterstrategy. Interestingly, this result is a disguised Farkas lemma, a fundamental theorem of linear algebra [27, Corollary 7.1h].

Lemma 3.2 *Let S be a CLPP and σ an admissible strategy in S. Then there is a counterstrategy τ of* MIN *such that $opt(S(\sigma, \tau)) = opt(S(\sigma))$.*

Proof. If $opt(S(\sigma)) = +\infty$, the claim is trivial, because $S(\sigma, \tau)$ has less constraints.

Otherwise, rewrite the linear program $\max(1^T x | S(\sigma))$ as $\max(1^T x | Ax \leq b)$. Since its optimal value M is finite, for each solution satisfying $Ax \leq b$, the inequality $1^T x \leq M$ holds.

By the affine form of Farkas Lemma (see, e.g., [27, Corollary 7.1h]), we have that $1^T x \leq M$ is a positive linear combination of some constraints in $Ax \leq b$.

Let n be the number of variables in $S(\sigma)$. Then there exists at most n linearly independent constraints, and since the solution is bounded there must exist at least one constraint per variable. Thus we are able to choose one constraint per variable that are linearly independent such that $1^T x \leq M$ is a positive linear combination of the chosen constraints. The new linear program $max(1^T x | A'x \leq b')$ has the optimal value M, because $1^T x \leq M$ and $A'x \leq b'$ is feasible. Now take the constraint selected for every compulsory variable as the strategy τ and $M = opt(S(\sigma, \tau))$. □

4 When Stability Implies Optimality

A strategy with no attractive switches is called *stable*. For many important subclasses of the CLPP, we can show that every stable strategy is optimal. As stated in Section 2.9, instances with this property can be solved in randomized subexponential time. The subexponential analysis of the algorithms, adapted from combinatorial linear programming, relies on the fact that once a local optimum of a subsystem (with a subset of the original constraints) has been found, we never need to return to a strategy in the subsystem. When stability does not imply optimality, this property no longer holds, and the algorithms might even fail to correctly find an optimum. The algorithms are explained in Section 5.

When investigating for which subclasses stability implies optimality, it is useful to distinguish between the following two major types of CLPPs.

Potentially Unbounded. These CLPP may potentially have *unbounded* optimal values. Such CLPPs arise, in particular, from mean payoff games (MPGs) and the longest-shortest paths (LSP) problem [10]. They are slightly more difficult to deal with. Duality theory for the LSP problem is treated in Section 4.4. We currently do not know whether deciding unboundedness of an arbitrary CLPP is in $\mathrm{NP} \cap \mathrm{coNP}$ (we can do it for CLPP instances corresponding to MPGs, and many others). Some necessary and sufficient conditions are investigated in Section 6.

Strongly Bounded. Such CLPPs are always guaranteed to have finite optimal values; moreover, any strategy choice by one of the players does not allow the other player to achieve an infinite value. We call such systems *strongly bounded*. They arise, e.g., from discounted payoff games (DPGs) and stopping simple stochastic games (SSGs). Although mean payoff games produce non-strongly bounded CLPPs, they are reducible to DPGs [29].

In Section 4.1 we show that in strongly bounded systems stability does indeed imply optimality. A number of other properties, including an important duality theorem, are also proved. Section 4.2 covers discounted systems, and shows that they are strongly bounded. Finally, in Section 4.3 we show that stability implies optimality for a class of CLPPs that cover MPGs and the LSP, but is not strongly bounded.

4.1 Strong Boundedness, Uniqueness, Duality

In this section we introduce the concept of a strongly bounded CLPP, by stipulating finiteness of target values in primal and dual instances, for any choice of admissible strategies by the players. We show that strongly bounded systems have a number of desirable properties. They belong to $\mathrm{NP} \cap \mathrm{coNP}$, and can be optimized in randomized subexponential time. Furthermore, the values of the primal and dual *coincide* (strong duality).

4.1.1 Strong Boundedness

This is a simple, yet very useful concept.

Definition 4.1 (Strong Boundedness) *Say that a CLPP instance S is* strongly bounded *if for every admissible strategies σ of* Max *in S and τ of* Min *in S^d one has: $opt(S(\sigma,\tau)) < +\infty$ and $opt(S^d(\sigma,\tau)) > -\infty$.* □

As a direct consequence of the definition and Lemma 6.2 below, we have the following nice property relating strong boundedness and nonsingularity.

Proposition 4.2 *For a strongly bounded CLPP instance, and any choice of admissible strategies by the players, the resulting constraint matrix is nonsingular.* □

Although strong boundedness may seem unrealistically restrictive, it nevertheless covers and immediately applies to stopping simple stochastic and discounted payoff games (see Section 2.3), since strong boundedness automatically holds for such games; see Section 4.2. Note that strong boundedness does not hold for CLPP instances acquired from mean payoff games, or the LSP problem.

4.1.2 Minimax Principle

We start with a useful general property of strongly bounded CLPPs, equating the optimal values in the primal and dual CLPPS, for any fixed pair of admissible strategies of the players.

Proposition 4.3 (Minimax Principle) *For every strongly bounded system S and any admissible strategies σ of* Max *in S and τ of* Min *in S^d one has $opt(S(\sigma,\tau)) = opt(S^d(\sigma,\tau))$.*

Proof. Assume $opt(S(\sigma,\tau)) = M \neq N = opt(S^d(\sigma,\tau))$. Since $M < +\infty$, by Lemma 3.1, an optimal solution u^* witnessing for M satisfies $S(\sigma,\tau)$ as equalities $S^=(\sigma,\tau)$, and $M = \sum_i u_i^*$. Similarly, v^* witnessing N satisfies $S^d(\sigma,\tau)$ as equalities $S^=(\sigma,\tau)$, and $N = \sum_i v_i^*$. Since u^* is different from v^*, $u^* + \alpha(u^* - v^*)$ is a solution to the system $S^=(\sigma,\tau)$ for *every* $\alpha \in \mathbb{R}$. Indeed, writing $S^=(\sigma,\tau)$ in the form $Ax = b$, it is easy to verify that $A(u^* + \alpha(u^* - v^*)) = b$. Note that $\sum_i (u_i^* - v_i^*) = M - N \neq 0$ by assumption.

Selecting a negative or positive α big enough we can make M arbitrarily big (pick any $\alpha > 0$ if the sum of coordinates in $u^* - v^*$ is positive and any $\alpha < 0$ otherwise). This contradicts to boundedness of M. Thus, $M = N$. □

4.1.3 Stability and Optimality for Strongly Bounded Systems

We are now ready to show that in strongly bounded systems, any stable strategy is also optimal. This property in itself motivates the definition and study of such systems, as it implies that they are efficiently solvable.

Lemma 4.4 *Let σ be stable in a strongly bounded S, and τ be an optimal response to σ (see Lemma 3.2). Then*

$$opt(S(\sigma)) = \max_{\sigma} opt(S(\sigma,\tau)),$$

i.e., MAX *cannot improve the value even if* MIN *invariably uses* τ.

Proof. The $\leq$ relation is obvious.

Suppose, MIN keeps his strategy τ, and in σ (that selects tight constraints) MAX makes a *flat* multiple switch (resulting in a new strategy σ') to other constraints satisfied as equalities in an optimal solution x^*. Lemma 4.5 demonstrates that a flat switch does not change the optimum value.

Assume, MIN keeps his strategy τ, but MAX makes an *anti-attractive switch*. Lemma 4.6 shows that such a switch does not improve the optimum value.

Any nonattractive multiple switch is a combination of a flat and an anti-attractive one, and the conclusion follows. □

Lemma 4.5 *Assuming strong boundedness of* S*, every flat switch for* MAX *in* $S(\sigma, \tau)$ *(only* MAX *has choices) preserves the optimal value. Symmetrically, every flat switch for* MIN *in* $S^d(\sigma, \tau)$ *(only* MIN *has choices) preserves the optimal value.*

Proof. By Lemma 3.1, $u^* = \arg\max(1^T x | S(\sigma, \tau))$ satisfies $S^=(\sigma, \tau)$. After making a flat switch, to another strategy σ', the vector u^* is still a solution to $S^=(\sigma', \tau)$. Again by Lemma 3.1[7], u^* is an optimal solution to $\max(1^T x | S(\sigma', \tau))$. □

Lemma 4.6 *An anti-attractive switch for* MAX *in* $S(\sigma, \tau)$ *(only* MAX *has choices) does not increase the optimal value. Symmetrically, an anti-attractive switch for* MIN *in* $S^d(\sigma, \tau)$ *(only* MIN *has choices) does not decrease the optimal value.*

Proof. By Lemma 3.1, $u^* = \arg\max(1^T x | S(\sigma, \tau))$ satisfies $S^=(\sigma, \tau)$. After making an anti-attractive switch, to another strategy σ', the vector u^* is still a *feasible* (but non-tight) solution to the dual system $S^d(\sigma', \tau)$, because such a switch corresponds to the following system of inequalities:

$$\begin{aligned} x &= p \text{ (in } \sigma), \\ &\vee \\ x &> p' \text{ (in } \sigma'). \end{aligned}$$

Therefore, the optimal solution to $S^d(\sigma', \tau)$ is smaller (in value) and the claim follows by Proposition 4.3. □

Corollary 4.7 *Every stable strategy in a strongly bounded CLPP instance is optimal.*

Proof. Suppose we have a stable strategy, but there is a better one. It can be obtained by a multiple switch. This multiple switch decomposes in two switches, one flat, and one anti-attractive. By the above lemmas, the first one preserves the value and the second one decreases it. □

[7]Here we essentially rely on strong boundedness. In general a switch to σ' may lead to unboundedness.

Corollary 4.7 will additionally be useful for proving Theorem 6.1.

Corollary 4.8 (Subexponential Optimization) *Every strongly bounded CLPP can be optimized in randomized subexponential time.*

Proof. See Section 5.1, where we discuss algorithms. □

4.1.4 Anti-Attractive Switches Needed in Unbounded Systems

Lemma 4.6 shows that in strongly bounded systems, anti-attractive switches cannot improve the value. This is no longer true in non-strongly bounded systems. Consider the system (only MAX has choices):

$$\begin{array}{lcllcl} x_1 & \leq & -3, & x_2 & \leq & -2, \\ x_1 & \leq & 2x_2, & x_2 & \leq & x_1, \end{array} \tag{2}$$

which is not strongly bounded. Let σ (stable) select the first constraints for $x_{1,2}$, resulting in a finite optimal value. Both switches for x_1 and x_2 look anti-attractive, but making them simultaneously leads to an unbounded improvement of the target function $\max(x_1 + x_2)$. This is depicted in Figure 1.

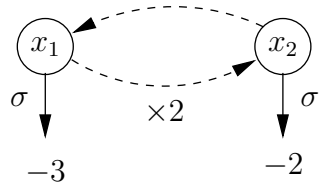

Figure 1: System (2) is shown in graph form. The current strategy σ leads directly to values -3 and -2. Both possible switches are anti-attractive, but making them both would create a loop which makes the system unbounded.

4.1.5 Weak Duality

We first show a weak form of duality for strongly bounded systems.

Lemma 4.9 (Weak Duality) *For every strongly bounded CLPP instance S and every pair of strategies σ, τ admissible in S, S^d one has:*

$$\max(1^T x | S(\sigma)) \leq \min(1^T x | S^d(\tau)). \tag{3}$$

Proof. We have the following chain of inequalities:

$$\max(1^T x | S(\sigma)) \leq \max(1^T x | S(\sigma, \tau)) = \min(1^T x | S^d(\sigma, \tau)) \leq \min(1^T x | S^d(\tau)),$$

where the first and third inequalities hold since for every σ, τ, the system $S(\sigma, \tau)$ is less constrained than $S(\sigma)$ and $S(\tau)$; the second equality holds by Proposition 4.3. □

4.1.6 Strong Boundedness is Essential for Weak Duality

The example of the non-strongly bounded system S, where MAX controls x:

$$\begin{array}{rclcrcl} x & \leq & -5, & \qquad & x & \leq & y, \\ y & \leq & x, & & y & \leq & 5 \end{array}$$

shows that the strong boundedness requirements in the statement of the Lemma 4.9 is essential. Indeed, when Max chooses $x \leq y$, then Min's best reply is $y \leq 5$, and the optimal solution is $x = y = 5$. But if in the dual system S^d Min chooses $y \geq x$, then Max's best reply will be $x \geq -5$, and the optimal solution is $x = y = -5$. But $5 \not\leq -5$. This example explains why strong boundedness is essential. In the proof we rely on Proposition 4.3 to conclude that whenever Max and Min stick to σ, τ, the values of the primal and dual systems coincide. In the unbounded case illustrated by the example this is false: when σ selects $x \leq y$ and τ selects $y \leq x$ (both admissible), then $\max(1^T x | S(\sigma, \tau)) = +\infty$, but $\min(1^T x | S^d(\sigma, \tau)) = -\infty$.

4.1.7 Strong Duality

Actually, $\leq$ in (3) can be strengthened to equality:

Theorem 4.10 (Strong Duality) *For every strongly bounded CLPP instance S*

$$\max_{\sigma} \max(1^T x | S(\sigma)) = \min_{\tau} \min(1^T x | S^d(\tau)). \tag{4}$$

Proof. Consider a sequence of strategies $\sigma_1, \tau_1, \sigma_2, \tau_2, \ldots, \sigma_p, \tau_p$ corresponding to any sequence of attractive switches for Max in S, where he selects better and better strategies σ_i, and each time Min responds with the best counterstrategy τ_i. The last pair σ_p, τ_p forms a stable pair of strategies. Neither a switch in σ_p can increase, nor in τ_p can decrease the value of $1^T x$. Now, let us start iterative improvement by attractive switches in the dual system S^d starting from the initial strategy τ_p. But this strategy is stable, and by Proposition 4.3 we know that $\max(1^T x | S(\sigma_p, \tau_p)) = \min(1^T x | S^d(\sigma_p, \tau_p))$. Since $\max_\sigma \max(1^T x | S(\sigma)) \geq \max(1^T x | S(\sigma_p, \tau_p)) = \min(1^T x | S^d(\sigma_p, \tau_p) \geq \min_\tau \min(1^T x | S^d(\tau))$, the conclusion follows by Lemma 4.9. □

4.2 Stability Implies Optimality for Discounted Systems

Discounted CLPPs can be seen as a generalization of discounted payoff and stopping simple stochastic games.

Definition 4.11 *In a* discounted CLPP *instance all constraints have form either $x \leq c$, for a constant c, or $x \leq \sum_i a_i \cdot x_i$, where x and x_i are variables, and a_i are positive constants with $\sum_i a_i < 1$.* □

Note that we can equivalently allow for only one type of constraints, namely: $x \leq \sum_i a_i \cdot x_i + w$, where $w \in \mathbb{R}$, because such a constraint is expressible via $x \leq \sum a_i x_i + a'x'$ and $x' \leq w/a'$, where $a' > 0$ is small enough.

Lemma 4.12 *Every discounted CLPP is strongly bounded.*

Proof. Suppose for some selection of strategies the resulting constraint system $Ax \leq b$ has solutions x^p with unbounded costs $1^T x^p = C_p$, where C_p form an unboundedly increasing sequence. It follows that for every p there is an i such that $x_i^p \geq C_p/n$, where n is the dimension of A. Since x^p satisfy the constraints, $x_i^p \leq \sum a_j x_j^p + b_i$, for the maximal component x_i^p of x^p. Hence $x_i^p \leq \delta x_i^p + b$, where $\delta < 1$ is the maximum sum of off-diagonal elements in A, and b is the largest right-hand side. Thus, $x_i^p \leq b/(1-\delta)$, contradicting, together with $C_p/n \leq x_i^p$, the unboundeness of C_p. □

Theorem 4.13 *In discounted CLPPs, every stable strategy is optimal.*

Proof. By Lemma 4.12 every discounted system is strongly bounded and the result follows from Corollary 4.7. □

In [3] we gave another, more complicated proof of this theorem (based on graph and hypergraph potential transformation, similar to the proof of Theorem 4.14 below), which is now a direct consequence of the results for strongly bounded systems.

4.3 Stability Implies Optimality for the Univariate, Unit Coefficient Case

Let us consider the case of "one variable with coefficient one" in the right-hand sides, i.e., all the constraints have the form $u \leq v + w$, where u, v are variables, and $w \in \mathbb{R}$. We also allow compulsory constraints of the form $t \leq 0$ (correspond to sinks in a directed graph built on variables with directed edges corresponding to inequalities). Systems of this kind arise from MPGs and LSP problems [10]. These systems are not always strongly bounded, so we have to use other means for showing that stability implies optimality.

For reasons, which become clear later, we need to stipulate an important

Zero-Loop-Free Condition. *In the directed graph determined by the constraints there are no* zero-weight cycles. *A (simple) cycle is a sequence of edges $u \stackrel{w}{\to} v$ corresponding to constraints $u \leq v + w$. As usual, the weight of this cycle is the sum of edge weights.* □

Imposing this condition does not lead to a loss of generality. One way to guarantee this condition for the LSP instances generated from MPGs is described in Section 2.3.1, when all 0-weight loops are transformed into negative-weighted ones (i.e., inattractive for Max). In other applications, when we want to figure out whether Max can secure a $+\infty$ distance to the sink, 0-weight loops giving $+\infty$ are good for Max. To turn such loops into positive, we make a dual transformation: multiply all edge weights by $n+1$ and add 1.

Theorem 4.14 *Let σ be a stable strategy, the graph determined by S does not have zero-weight cycles, $S(\sigma)$ is feasible and has a finite*[8] *optimal solution. Then σ is optimal.*

Proof. Suppose σ is a stable discrete strategy. Consider the *potential* q defined on variables by giving them the respective values of an optimal solution. For every edge/constraint $u \leq v + w$ in $S(\sigma)$ the potentials satisfy $q(u) \leq q(v) + w$, with equalities for the tight constraints (e.g., for those selected in σ.) Define an optimal counterstrategy $\tau(\sigma)$ for MIN as any selection of exactly one tight constraint per his variable. Make a *potential transformation* of the graph of system S (of the whole system S, rather than just of $S(\sigma)$), by changing the weight of each edge according to $w'_{u,v} = w_{uv} - q(u) + q(v)$. By stability of σ, for MAX edges not in σ we have the opposite inequality $q(u) \geq q(v) + w$. After this transformation: 1) all edges in σ become 0-weight, and all other MAX edges become non-positive (by stability, as explained above); 2) all edges in $\tau(\sigma)$ become 0-weight, and all other edges of MIN become nonnegative; 3) weights of all loops remain *unchanged*; 4) weights of finite paths $v_0, \ldots, v_l$ change by a constant $-q(v_0)+q(v_l)$ (telescoping), hence the relation $(<, =, >)$ between costs of two paths to a sink remains unchanged.

Let us assume, toward a contradiction, that σ is not optimal and there exists a better strategy σ', providing for a larger value of the target function, hence of some variable v_0. Suppose, MAX switches to this strategy σ', but MIN persistently uses the strategy $\tau(\sigma)$. Now in the trace created from v_0 uniquely determined by the pair of strategies σ' and $\tau(\sigma)$ every edge is *nonpositive* (we assume the potential transformation has been performed). This trace may be:

1. either *finite*, terminating in a sink $t \leq 0$ in this case the value for v_0 is non-positive, and this contradicts to the assumption that σ' provides a better value for v_0 than σ;

2. or *infinite*; in this case the value for v_0 must be *negative*; indeed, due to our construction, the graph does not contain zero-weight cycles, and this property is preserved during the potential transformation. All edges used by σ' and $\tau(\sigma)$ have nonpositive weights after the transformation. Since there are no zero-weight cycles, the only possible cycles formed by σ' and $\tau(\sigma)$ are negative. Hence v_0 will be a negative value (or $-\infty$) in this case.

This contradiction shows that σ is indeed an optimal strategy. □

In [3] we gave similar proofs, based on potential transformations, for discounted univariate and multivariate cases. Now these cases are covered by simpler proofs based on duality; see Section 4.2.

4.3.1 Zero-Loop-Free Condition is Essential

The proof above shows that the Condition is essential. Indeed, if there are zero-weight loops, there may be stable but non-optimal strategies. Consider the following simple example.

[8]This assumption is no loss of generality; see the end of Section 2.7 on eliminating unbounded variables.

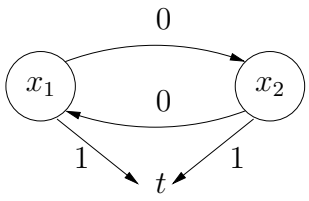

Figure 2: In this controlled LP there is a strategy that is stable, but not optimal. Round vertices are controlled.

The corresponding system of constraints is:

$$\begin{cases} x_1 \leq 1+t, \\ x_1 \leq x_2, \end{cases} \qquad \begin{cases} x_2 \leq 1+t, \\ x_2 \leq x_1, \end{cases} \qquad t \leq 0.$$

The strategy choosing the first constraints for each of x_1, x_2 is stable, and gives finite values 1, 1 for both. However, a better strategy consists in selecting the second constraints, which leads to an unbounded solution, $x_1 = x_2 =$ anything. Indeed, by deviating from the first stable strategy Max creates a 0-loop, which allows for unboundedness.

4.4 LSP-Duality

CLPP instances arising form the LSP problem are not strongly bounded, in general. This case deserves and requires a separate treatment.

4.4.1 LSP on DAGs

For the LSP on DAGs we have *strong duality* (4) (both sides of the equality are always defined and finite), because for every DAG the constraint system S is *strongly bounded*, and our Theorem 4.10 applies. Consequently, we can proceed by attractive switches for Max or for Min. Either way the iterative improvement stops after polynomially many steps, which can be shown by induction on a topological sorting of the DAG vertices. In the former case at every iteration an optimal counterstrategy for Min is computable via the shortest paths forest (from each vertex to the sinks). In the latter case at every iteration one needs to compute the *longest paths* forest, which reduces to shortest paths by negating edge weights (note that this version of longest/shortest paths is not NP-hard).

4.4.2 LSP on Graphs Obtained from Ergodic MPGs

Ergodic MPGs (all game vertices have the *same value*) form a reduction class [18, 10], and we consider them here for simplicity. In [10] we obtained an LSP-instance S_1 (primal CLPP-instance) from any MPG by adding a 0-weight edge leading to a 0-weight self-loop (retreat) from every vertex of Max (special care should also be taken to avoid 0-weight loops in the original MPG). After that the 0-weight self-loop is turned into a sink and we show that Max can secure the value > 0 in the initial MPG iff the resulting LSP instance S_1 is

unbounded, i.e., $\max_\sigma \max(1^T x | S_1(\sigma)) = +\infty$. The iterative improvement for S_1 described in [10] terminates either with a finite optimal value, or reports unboundedness.

Symmetrically, we can add 0-retreat edges to every MIN vertex, turn the retreat self-loop into the sink, thus forming a dual CLPP-instance S_2 (of Shortest-Longest Paths, SLP for short), and apply the dual iterative improvement for MIN strategies in S_2, starting from the "retreat everywhere". By the symmetric argument, MIN can secure a value < 0 in the initial MPG iff in the dual SLP-instance S_2 MIN can achieve an unbounded value, i.e., $\min_\tau \min(1^T x | S_2(\tau)) = -\infty$.

Since zero-weight loops in the original MPG can be avoided by a simple transformation [10], we have the LSP/SLP-duality in the following form (note that S_1 and S_2 are not duals of each other).

Theorem 4.15 *For every ergodic MPG without zero-weight loops and the corresponding LSP- and SLP-instances S_1 and S_2 obtained by adding retreats/sinks as described above, exactly one of*

1. $\max_\sigma \max(1^T x | S_1(\sigma))$, *or*
2. $\min_\tau \min(1^T x | S_2(\tau))$

is unbounded ($+\infty$ or $-\infty$, resp.) □

Corresponding to cases 1 and 2, either player MAX secures a value > 0 or player MIN secures a value < 0 from every vertex of the initial game. NP ∩ coNP-membership follows immediately; actually, our result from [10] is stronger, suggesting a subexponential randomized decision algorithm. Note that when performing the strategy improvement for MIN, every iteration requires solving an instance of the *longest paths* problem, which is not NP-hard and reduces to shortest paths by negating the edge weights.

5 Algorithms

In this section we consider discrete (simplex-like) and interior point local search algorithms for the CLPP.

5.1 Discrete Switching Algorithms

For all CLPPs, attractive switches are profitable (Theorem 2.4). When the additional property that stability implies optimality holds, the following generic algorithm correctly finds a global optimum. Start with any admissible strategy and make attractive switches until a stable strategy is found. Specific algorithms are obtained by specifying how to find the initial strategy and how to select one or many attractive switches to make.

5.1.1 Single and Multiple Switch Algorithms

We briefly describe several switching schemes that can be employed in the generic algorithm. For more details and applications see [6, 11]. Unfortunately, none of them is known to be better-than-exponential [6].

- *Single Random Switch*: select uniformly at random exactly one attractive switch in each iteration.
- *Single Greedy Switch*: select "greedily" exactly one of the most attractive switch (switch to the most relaxing constraint) in each iteration.
- *All Attractive Switches*: perform all attractive switches at every iteration.
- *Random Multiple Attractive Switches*: flips a coin for every attractive switch in each iteration (or uniformly at random choose a subset of all available attractive switches.)

5.1.2 Randomized Subexponential Algorithms

Khachiyan was the first to show that linear programming can be solved in polynomial time. His algorithm, as well as the one of Karmarkar, are polynomial in the size of the input, but the running times cannot be bounded solely in the numbers of constraints and variables. The question whether there is an algorithm that is (strongly) polynomial in the combinatorial model of computation remains open, and research in this area continues. In a major breakthrough, Kalai showed in 1992 that the problem does at least have a strongly subexponential solution [20]. Matoušek, Sharir, and Welzl independently proved the same result for another algorithm [23]. In 1995, Ludwig was the first to propose using these techniques in a game-theoretic setting [22], but in a restricted form, only for games on graphs of outdegree 2. Ludwig [22] suggested a subexponential $2^{O(\sqrt{n})}$ (n is the number of vertices) algorithm for *binary* simple stochastic games, in which every vertex has outdegree *at most two*. Although a general simple stochastic game, with *arbitrary* vertex outdegrees, reduces to a binary one,

this leads to a *quadratic* $O(n^2)$ blow-up in the number of vertices, and Ludwig's algorithm, unfortunately, becomes *exponential*: $2^{O(\sqrt{n^2})} = 2^{O(n)}$. One may argue that Ludwig's algorithm is anyway *subexponential in the length of input*, since the length of input in this case is $O(n^2)$ (for representing edges). However, a straightforward brute-force algorithm, which systematically considers all up to $2^{n \log n}$ pure strategies and selects the best one, is also *subexponential* in the length of input in this case! So, Ludwig's algorithm does not show drastic advantages over the brute-force one. Therefore, a more adequate meaning of *subexponential* in this context should be given in the *number of vertices*, rather than the input size.

The first (really) subexponential algorithms $2^{O(\sqrt{n \log n})}$ in the number of vertices n for simple stochastic games of arbitrary outdegree are based on different than Ludwig's randomization schemes and were first suggested in [8, 7, 6, 4, 11].

Kalai-Style Randomization for Controlled LP. Kalai's approach can be successfully adapted for solving the CLPP in subexponential time (in the number of variables and constraints).

Given an instance S of CLPP, a *subproblem* F is obtained by fixing one optional constraint for some controlled variable. In combinatorial terms, F is a *facet* in the space of positional strategies of the controller in S. This establishes the "constraints $\leftrightarrow$ facets" relation.

The Kalai-style algorithm for the CLPP proceeds as follows (cf., [7, 9]). Given an instance S and a strategy σ, it applies itself recursively to find a big enough set $\mathcal{S}$ of *better* subproblems (facets), which have optimal strategies better than σ (with a larger value of the target function). This is accomplished by starting with the subproblem where all optional constraints, except those in σ, are removed, and iteratively adding improving (attractive) constraints; each such constraint defines a better subproblem. (The algorithm may terminate if during this phase it finds an optimal strategy in S, or $+\infty$ results.) Sampling uniformly at random from $\mathcal{S}$, the algorithm selects a subproblem F, in which it recursively finds an optimal strategy σ'. All facets in $\mathcal{S}$ are ordered according to the values of their best strategies. Due to randomization, all positions for F in this ordering are equiprobable. The algorithm never again visits a facet worse than F. It follows that the *hidden dimension* (the unknown number of better subproblems remaining after σ' is obtained) decreases randomly. This results in a randomized algorithm with subexponential $2^{O(\log m \cdot \sqrt{n/\log n})}$ expected number of iterations for the CLPP (whenever stability implies optimality), where n, m are the numbers of controlled variables and optional constraints. If we assume that m is polynomially bounded in n, the bound is $2^{O(\sqrt{n \log n})}$.

Matoušek-Sharir-Welzl-Style Randomization for Controlled LP. Associating optional constraints with facets in the space of the controller's positional strategies, provides the intuitions necessary for applying other approaches from combinatorial linear programming to the CLPP. In particular, we succeeded to adapt the algorithm by Matoušek, Sharir, and Welzl (MSW). This algorithm is more intuitive, and easier to explain for CLPP than Kalai's. The function $opt\,(S(\sigma))$ works as the target function being optimized.

The SWITCH function takes a strategy σ and an optional constraint $x \leq e$ and returns the strategy obtained by changing the choice of σ for x to $x \leq e$. In combinatorial terms, line 3 of the algorithm selects a facet in the space of positional strategies, while line 4 recursively

optimizes on the rest of this space. The condition in line 6 can be efficiently checked using the switch attractiveness.

Algorithm 1: MSW-Style Algorithm for CLPP
CLPP-MSW(S, σ_0)
(1) **if** every controlled variable has only one constraint in S
(2) **return** σ_0
(3) choose a random optional constraint $x \leq e$ not in σ_0
(4) $\sigma^* \leftarrow$ CLPP-MSW$(S \setminus \{x \leq e\}, \sigma_0)$
(5) $\sigma_1 \leftarrow$ SWITCH$(\sigma^*, x \leq e)$
(6) **if** *opt* $S(\sigma_1) >$ *opt* $S(\sigma^*)$
(7) $S' \leftarrow S \setminus \{x \leq e' : e' \neq e\}$
(8) **return** CLPP-MSW(S', σ_1)
(9) **else return** σ^*

The number of iterations of this algorithm is $2^{O(\sqrt{n \log(m/\sqrt{n})} + \log m)}$, similar to Kalai's above. Again, if m is polynomial in n, the bound is $2^{O(\sqrt{n \log n})}$.

5.2 Interior Point Methods

The previous discrete switching algorithms can be seen as simplex-like, visiting the vertices of the Cartesian product $\prod_{i=1}^{n} S_i$ (the space of pure/discrete strategies), monotonically improving the target function value. Let us extend this by allowing the algorithm to visit the *interior* of the product of simplices. The adequate setting for this is as follows (previously suggested and investigated in the context of simple stochastic games in [25, 24]).

For every controlled vertex x_i let a nonnegative vector $\bar{\lambda}_i = (\lambda_i^1, \ldots, \lambda_i^{n_i})$ have the l_1-norm equal to 1, i.e., $\sum_{j=1}^{n_i} \lambda_i^j = 1$. Combine all the constraints for each controlled variable x_i into a weighted sum:

$$\sum_{j=1}^{n_i} \lambda_i^j (p_i^j + w_i^j). \tag{5}$$

Collect all the constraints (5), for all controlled variables, together with all compulsory constraints, and denote the resulting system by $S(\lambda) = S(\bar{\lambda}_1, \ldots, \bar{\lambda}_n)$. Call the vector of vectors $\lambda = (\bar{\lambda}_1, \ldots, \bar{\lambda}_n)$ *a mixed strategy*. Each such mixed strategy is an interior point of the product of simplices. In contrast, a strategy in which every $\bar{\lambda}_i$ has exactly one component λ_i^j equal 1, is called *pure*. Call a mixed strategy λ *admissible* if the system of constraints $S(\lambda)$ is feasible.

Similar to the discrete case, we will introduce a notion of attractive interior point switch, and will prove two facts, guaranteeing the correctness of a wide range of interior point algorithms (e.g., the generic algorithm from Section 5.1 will work fine with interior point switches):

1. for an admissible strategy, any combination of attractive switches is profitable and leads to an admissible strategy (Section 5.2.1);

2. for any stable mixed strategy (no attractive switches) there is a pure no-worse stable strategy;

5.2.1 Attractiveness and Profitability for Interior Point Switches, Discretization

Definition 5.1 *Given an admissible interior point strategy $\lambda = (\bar{\lambda}_1, \bar{\lambda}_2, ..., \bar{\lambda}_n)$, let x^* be an optimal solution to the system $S(\lambda)$. Then an interior point (single) switch for variable x_i from $\bar{\lambda}_i$ to $\bar{\lambda}'_i$ is called* attractive*, if*

$$\sum_j \lambda_i^j (p_i^j(x^*) + w_i^j) < \sum_j \lambda_i'^j (p_i^j(x^*) + w_i^j). \tag{6}$$

A switch from λ to another admissible strategy $\lambda' = (\bar{\lambda}'_1, \ldots, \bar{\lambda}'_n)$ is profitable*, if for each variable v one has*

$$\max\{v \textit{ subject to } S(\lambda)\} \;\leq\; \max\{v \textit{ subject to } S(\lambda')\},$$

and for some variable the inequality is strict*.* □

Theorem 2.4 is a special case of the following general result.

Theorem 5.2 *Any combination of interior point attractive switches is profitable.*

Proof. Assume attractive switches were made, from strategy λ to λ'. Let x^* be the optimal solution to the old LP $S(\lambda)$. For every switched variable x_i, we have the old constraint $x_i \leq \sum_j \lambda_i^j (p_i^j + w_i^j)$, and the new constraint $x_i \leq \sum_j \lambda_i'^j (p_i^j + w_i^j)$. By definition of an attractive switch, we have (6) satisfied.

It is easy to see that x^* is a solution to the new LP $S(\lambda')$, because every constraint is either the same or more generous, i.e., has a larger right-hand-side, by (6).

Take one switched variable x_i, let $x'_i = x_i^* + \varepsilon$ for a small enough $\varepsilon > 0$, and let $x'_j = x_j^*$ for $j \neq i$. Then $x'$is a solution to the new LP $S(\lambda)$. This follows from the two facts:

1. x'_i does not violate its constraint because the attractive switch made it more generous.

2. The increase of x_i does not offend any other constraints, because the polynomials p_i^j have only non-negative coefficients, so no constraint is tightened by an increase in x_i.

We have thus shown that the resulting strategy is admissible and that there exists a better solution to the new LP. Therefore, the switch is profitable. In addition, we see that the values of variables where attractive switches are made are guaranteed to increase in an optimal solution to the new system.

Note that the argument above essentially uses the fact that the polynomials p_i^j in right-hand sides of constraints have nonnegative coefficients. The above argument fails, if negative coefficients are allowed. □

Discretization. Every stable mixed (interior-point) strategy can be made discrete (pure), without deteriorating the target value. This simplification allows us to concentrate on discrete/pure strategies while proving optimality of stable strategies, as in Section 4.

Theorem 5.3 *For every mixed stable strategy there exists a discrete/pure strategy providing for a no-worse value of the target function.*

Proof. Assume $\lambda = (\bar{\lambda}_1, \ldots, \bar{\lambda}_n)$ is mixed and stable, and the optimal solution to the system $S(\lambda)$ is x^*. For every non-discrete $\bar{\lambda}_i = (\lambda_i^1, \ldots, \lambda_i^{n_i})$ consider all $\lambda_i^j > 0$. For all such λ_i^j the values of the corresponding right-hand sides of the constraints $p_i^j(x^*) + w_i^j$ are the same, since there are no attractive switches. Denote this value z.

Obtain $\bar{\lambda}_i'$ from $\bar{\lambda}_i$ by discretization as follows. Take any k such that $\lambda_i^k > 0$, set $\lambda_i'^k$ equal one, and all other components of $\bar{\lambda}_i'$ equal zero. Since $x_i^* \leq \sum_{j=1}^{n_i} \lambda_i^j (p_i^j(x^*) + w_i^j) = \sum_{j=1}^{n_i} \lambda_i^j z = 1 \cdot z = p_i^k(x^*) + w_i^k$, switching from $\bar{\lambda}_i$ to $\bar{\lambda}_i'$ leaves the solution x^* feasible. Therefore, the value of the target function does not get worse. Repeat the same argument for each non-discrete component $\bar{\lambda}_i$. □

In [3] we show that after discretization described in the proof of the theorem above, the resulting strategy may become *unstable*. In this case it will be enough to proceed with attractive switches, and apply the argument again. This will eventually result in a discrete stable strategy.

6 Optimality Conditions

In many cases, in particular, when translating MPGs to CLPPs, the most important issue we are interested in is boundedness. We want to know if MIN has a strategy such that for every strategy of MAX, the resulting system is either bounded or infeasible. In general, the importance of boundedness is explained by the following

Theorem 6.1 (Sufficient Optimality Condition) *Let σ, τ be a pair of stable strategies in a CLPP instance S. If $\max_{\sigma'} opt(S(\sigma', \tau)) < +\infty$, then σ is an optimal strategy in $S(\tau)$.*

Proof. The system $S(\tau)$ is strongly bounded, and Corollary 4.7 applies. □

The simplicity of this proof illustrates the usefulness of the strong boundedness concept. Stable strategies are local optima found by any of our iterative improvement algorithms. Not all such local optima are global, unless boundedness is guaranteed. For this reason, it is important to identify the characteristics of bounded systems. In this section we investigate necessary and sufficient conditions for boundedness of optimal values. Interestingly, boundedness is related to well-known linear algebraic concepts, like nonsingularity and kernels of linear operators.

6.1 Nonsingularity: A Necessary Condition of Boundedness

It turns out that player MIN, in order to be able to keep the optimal value of the CLPP instance *finite*, has to select counterstrategies in such a way that the resulting constraint matrix is *nonsingular*.

Lemma 6.2 *Suppose σ is admissible in S and $opt(S(\sigma)) < +\infty$. Then for every strategy τ of* MIN *for which $opt(S(\sigma, \tau)) < +\infty$, the matrix A of the constraint system $S(\sigma, \tau)$, written as $Ax \leq b$, is nonsingular.*

Proof. Let $x^* = \arg\max(1^T x | Ax \leq b)$. Such a vector exists by admissibility of σ. Assume, toward a contradiction, that A is singular. Then there is a nontrivial solution $y^* \neq 0$ to $Ax = 0$. Suppose $1^T y^* \neq 0$. Then $Ax^* + \lambda A y^* = A(x^* + \lambda y^*) \leq b$ for any $\lambda \in \mathbb{R}$. This contradicts the assumption that x^* is optimal. Thus $1^T y^* = 0$.

Let m be the number of variables in x, and let $I \subseteq \{1, \ldots, m\}$ be the set of indices such that $y_i^* \leq 0$ for all $i \in I$ and $y_j^* > 0$ for all $j \notin I$. Notice that I is a nontrivial subset of $\{1, \ldots, m\}$. Now construct y' in the following way. If $i \in I$, then set $y'_i = 0$, otherwise $y'_i = y_i^*$. Consider a constraint from $Ax \leq 0$. It has the form $x_i \leq p(x)$, where p is a homogeneous polynomial with nonnegative coefficients. We know that $Ay^* \leq 0$. If $i \notin I$, then $y'_i = y_i^*$ and $p(y') \geq p(y^*)$, so y' satisfies the constraint. On the other hand, if $i \in I$, then $y'_i = 0$, and since all entries in y' are nonnegative, $p(y') \geq 0$, so the constraint is satisfied. Thus $Ay' \leq 0$.

This means that $Ax^* + Ay' = A(x^* + y') \leq b$ and $1^T y' > 0$, thus $1^T(x^* + y') > 1^T x^*$, which contradicts the assumption that x^* is optimal. Thus 0 is the unique solution to $Ax = 0$ and A is nonsingular. □

Remark. Note that the special structure of CLPP (positiveness of constraints) is essentially used in the proof above. Indeed, consider the 2x2 matrix A with all entries equal 1, which *does not correspond to any CLPP instance*. Although $\max(1^T x | Ax \leq 0) = 0$, the matrix A is singular. For an arbitrary such (non-CLPP-generated) matrix we can only claim that its kernel is a linear subspace of the hyperplane $\{x | 1^T x = 0\}$, once the optimum is finite. □

6.2 Bounded Systems Have Unique Solutions

In Section 2.7 we defined attractive switches for a strategy σ as those that seem to give an improvement with respect to the value vector $x^* = \arg\max(1^T x | S(\sigma))$. In the definition the optimal solution was implicitly assumed to be unique. We are now ready to justify this assumption.

Corollary 6.3 *Suppose σ is admissible in S and $opt(S(\sigma)) < +\infty$. Then* $\arg\max(1^T x | S(\sigma))$ *is* uniquely *defined.*

Proof. Let τ be an optimal counterstrategy to σ. Rewrite $S(\sigma, \tau)$ as $Ax \leq b$. If x^* is an optimal solution to $Ax \leq b$, then we must have $Ax^* = b$ by Lemma 3.1, and since A is nonsingular by Lemma 6.2, this solution is unique. □

6.3 Nonsingularity Does not Imply Boundedness

An attempt to invert Lemma 6.2 fails, as shows the following

Proposition 6.4 *Nonsingularity of the constraint matrix (resulting from fixed player strategies) does not imply boundedness.*

Proof. Let strategies of Max, Min result in the constraints $x_1 \leq x_2$, $x_2 \leq 2x_1$. The matrix of this system is nonsingular:

$$A = \begin{pmatrix} 1 & -1 \\ -2 & 1 \end{pmatrix}, \; A^{-1} = \begin{pmatrix} -1 & -1 \\ -2 & -1 \end{pmatrix},$$

and the unique tight solution to the system is $x_1, x_2 = 0$. But $Ax \leq b$ has arbitrarily large solutions, e.g., $(1, 1)$, $(2, 2)$, $(3, 3)$, ..., none of which satisfies both constraints as equalities. □

6.4 Sufficient Condition for Boundedness

Nonsingularity is only a necessary condition. Combining it with an additional requirement gives sufficiency.

Lemma 6.5 *Suppose for every pair of strategies σ, τ of* Max, Min *the matrix A for the constraint system $S(\sigma, \tau)$, written in the form $Ax \leq b$, satisfies:*

1. *A is nonsingular;*

2. *in the inverse matrix A^{-1} every column sums up to a nonnegative number.*

Then $\max_\sigma\ opt(S(\sigma)) < +\infty$.

Proof. Consider the set of all solutions to the system $Ax \leq b$, which may be equivalently rewritten as $Ax + y = b$ with $y \geq 0$ for fresh variables y (i.e., y is a nonnegative vector of slack variables). Rewriting it equivalently as

$$x = -A^{-1}y + A^{-1}b, \tag{7}$$

we see that for every nonnegative vector $y^* \geq 0$ the solution x^* to (7) is *uniquely* determined. Now the question reduces to whether

$$\max(1^T x | x = -A^{-1}y,\ y \geq 0) < +\infty \tag{8}$$

(Note that we drop out the constant vector $A^{-1}b$ as nonsignificant for unboundedness.)

The question (8) is easy to answer. $-A^{-1}$ is a nonsingular linear operator transforming a nonnegative orthant (cone) $y \geq 0$ to another unbounded full dimensional cone with 0 as origin. The question (8) therefore becomes:

whether following any of the rays $-A^{-1}\lambda e_i$, for $\lambda > 0$ and e_i the i-th basis vector (1 in i-th position, zeros elsewhere) allows to unboundedly increase the sum of all coordinates?

It holds iff there is a column i in A^{-1} with a negative sum of elements, and the claim follows. □

6.5 Rational Systems

For CLPP instances in which all coefficients and constants are rational numbers, we can derive stronger characterizations of boundedness. The rationality assumption is reasonable in most practical circumstances; for instance, it holds for all CLPPs derived from parity, mean payoff, discounted payoff and simple stochastic games (when discounting factors and probabilities are rational).

Lemma 6.6 *For a rational feasible system $Ax \leq b$ the following conditions are equivalent:*

1. $\max(1^T x | Ax \leq b) < +\infty$,

2. $\forall y(Ay \leq 0 \Rightarrow 1^T y \leq 0)$,

3. *The system $Ay \leq 0$, $1^T y > 0$ is infeasible.*

Proof. (1 $\Rightarrow$ 2) Assume (1) and suppose, towards a contradiction, that $1^T y' = c > 0$ for some y' satisfying $Ay' \leq 0$. Take any feasible solution x' to $Ax \leq b$ (here we rely on feasibility) and any $\lambda > 0$. Then $x' + \lambda y'$ is feasible for $Ax \leq b$ with the target value larger than $\lambda c > 0$, which can be made arbitrarily large, a contradiction with (1).

(2 ⇒ 1) Assume (2) and suppose, towards a contradiction, that $\max(1^T x | Ax \leq b)$ is unbounded. Then there is an infinite sequence s of integer[9] vectors $Ax_1, Ax_2, \ldots$ such that $Ax_i \leq b$ and $1^T x_i < 1^T x_{i+1}$.

Select an infinite subsequence s_1 of s of vectors with non-increasing first coordinate. Similarly, for $k < n$, select an infinite subsequence s_{k+1} of s_k with non-increasing $k+1$-st coordinate. In the sequence s_n there are two elements $Ax_i \geq Ax_j$ with $i < j$. Consider $x_j - x_i$. By construction, $A(x_j - x_i) \leq 0$, but $1^T(x_j - x_i) > 0$, a contradiction with (2). Finally, (2) is obviously equivalent to (3). □

Remark. The feasibility assumption is essential. Consider the infeasible system $\{x \leq y - 1, y \leq x - 1\}$. Then 1 holds, but 2 fails. □

Combining Lemma 6.6 with the monotonicity of CLPP constraints, we obtain a stronger characterization of bounded systems, which gives an alternative way to determine whether a CLPP instance is bounded.

Corollary 6.7 *Let S be a rational CLPP instance, σ and τ strategies of* Max *and* Min*, such that $S(\sigma, \tau)$ written as $Ax \leq b$ is feasible. Then the following conditions are equivalent:*

1. $\max(1^T x | Ax \leq b) < +\infty$,
2. *0 is the unique optimal solution to* $\max(1^T x | Ax \leq 0)$

Moreover, both conditions imply that the matrix A is nonsingular.

Proof. (1 ⇒ 2) follows from Lemma 6.2 and (2 ⇒ 1) from Lemma 6.6. The last claim is implied by (2), since 0 is the unique solution to $Ax = 0$. □

[9]We may assume that A and b have integer components, by multiplying every constraint by an appropriate integer (least common multiple of denominators). Since $1^T x$ is unbounded on $Ax \leq b$, the feasible region contains integral points x with arbitrarily large values $1^T x$. Since A is integral, it takes integral vector values on such x.

7 Membership in NP ∩ coNP

Optimality of stable strategies is essential for the correctness of the algorithms from Section 5. It also directly implies NP ∩ coNP-membership;[10] There are, however, subclasses of the CLPP for which the "stable ⇒ optimal" condition does not hold, but other methods can be used to prove coNP-membership. This section investigates two such classes. The first one contains all CLPP instances with only positive *integers* as coefficients, while the second one contains instances with at most one variable in the right-hand side of every constraint.

7.1 Systems with Integer Coefficients

In this section we prove

Theorem 7.1 *The CLPP with nonnegative integer coefficients is in NP ∩ coNP.*

Throughout this section, we assume that the systems involved have at least one admissible strategy, and finite optimal solutions. This is no loss of generality, as it can be ensured by the addition of the so-called *retreat constraints* as follows. First compute an upper bound M for the largest optimal value the system can have *if it is bounded*. Then, for every controlled variable x, add the retreat constraint $x \leq -M - 1$. Symmetrically, for every uncontrolled variable y, add the constraint $y \leq M + 1$; see [3, 10, 12]. To prove Theorem 7.1 we will need a series of definitions and technical lemmas. Recall that by $S(\sigma)$, $S(\tau)$, and $S(\sigma, \tau)$ we denote the constraint system S with, respectively, all MAX constraints not in σ thrown away, all MIN constraints not in τ thrown away, all constraints not in σ, τ thrown away.

Definition 7.2 *Say that a system $S(\tau)$ has a* cyclic constraint *over variable x if there is a* MAX *strategy σ forcing a cyclic sequence of controlled variables $x = x_1, x_2, \ldots, x_k = x$ in $S(\sigma, \tau)$, which has a constraint $x_i \leq p_i(\overline{x}, x_{i+1}) + w_i$ for each $i \in \{1, 2, \ldots, k-1\}$. If there is a strategy with at least two* distinct *such sequences, then say that $S(\tau)$ has* multiple cyclic constraints *over x.* □

Lemma 7.3 *If $S(\sigma, \tau)$ has a cyclic constraint over some variable, then $S(\sigma, \tau)$ is either infeasible or unbounded.*

Proof. If $S(\sigma, \tau)$ is feasible and has a single cyclic constraint over x, then, by combining constraints, we get $x \leq c \cdot x + p(\overline{x} \setminus \{x\}) + w$. For any solution to $S(\sigma, \tau)$, the polynomial $p(\overline{x} \setminus \{x\})$ evaluates to some finite number. Since c is a positive integer constant, the constraint either holds for an infinite number of *nonnegative* values of x (if $c > 1$ or $c = 1$ and $p(\overline{x} \setminus \{x\}) + w \geq 0$, or for *no* values of x (if $c = 1$ and $p(\overline{x} \setminus \{x\}) + w < 0$). It is easy to see that if $S(\sigma, \tau)$ has a multiple cyclic constraint over x, it is similar to the single cyclic constraint case where $c > 1$, thus yielding unbounded values. □

[10] Here we refer to the *decision* version of the CLPP: given an instance S and a rational number k, is the optimal solution to S larger than k? See Section 2.9. Note that the unrestricted CLPP is always in NP.

The next lemma follows from the previous one and our assumptions.

Lemma 7.4 *Given a CLPP instance S, let σ be an optimal strategy for* MAX. *Then there is a* MIN*-strategy τ, such that $S(\sigma, \tau)$ has no cyclic constraints.*

Proof. Let x^* be an optimal solution to $S(\sigma)$. Since there are retreats and σ is optimal, $S(\sigma)$ has a feasible solution and x^* assigns every variable a finite value. If, for every choice of τ from the tight compulsory constraints, there exists a cyclic dependency, then by Lemma 7.3, the system is either infeasible or unbounded. Either way, we have a contradiction. □

Lemma 7.5 *Assume that*

1. *$S(\tau)$ is a system without multiple cyclic constraints or cyclic constraints with coefficients larger than 1,*
2. *σ is a stable strategy in $S(\tau)$ and $opt(S(\sigma, \tau))$ is finite, and*
3. *σ' is a strategy enforcing a cyclic constraint sequence $x_0, x_1, \ldots, x_k = x_0$ yielding unbounded values, which differs from σ* only *in variables in $\{x_1, \ldots, x_k\}$.*

If we make a potential transformation on $S(\tau)$, using the value of each variable under σ as the potentials[11]*, then every constraint for a variable $x_i, i \in \{1, \ldots, k\}$ in σ' becomes homogeneous.*

Proof. Let x^* be the optimal solution to $S(\sigma, \tau)$. For each x_i, let $x_i \leq c_i$ be its constraint under σ'. Further, let $r_i = c_i(x^*) - x^*_{i+1}$.

Consider variable x_1. Since σ is stable, we have

$$x^*_1 \geq r_1 + x^*_2 \geq r_1 + r_2 + x^*_3 \geq \cdots \geq r_1 + \cdots + r_k + x^*_1,$$

or equivalently, $0 \geq r_1 + \cdots + r_k$. Actually, for $S(\sigma', \tau)$ to be feasible, we must have $0 = r_1 + \cdots + r_k$.

Let w_1 be the constant term in c_1. After the potential transformation, the constant term will be $w'_1 = -x^*_1 + r_1 + x^*_2 = \cdots = -x^*_1 + r_1 + \cdots + r_k + x^*_1 = 0$. Similar reasoning shows that the constraints for variables $\{x_2, \ldots, x_k\}$ in $S(\sigma', \tau)$ will also become homogeneous after the potential transformation. □

Lemma 7.6 *Assume the same situation as in Lemma 7.5, but by switching to σ'* MAX *enforces the cyclic constraint and creates an infeasible system. Then, after the same potential transformation, at least one constraint for a variable $x_i, i \in \{1, \ldots, k\}$ in σ' has a negative constant term.*

Proof. Follows by an argument similar to the proof of Lemma 7.5. □

[11] See the proof of Theorem 4.14.

Lemmas 7.5 and 7.6 show that any single cyclic constraint, with coefficients equal to one, such that MAX can achieve unbounded values by only switching to the constraints associated with the cyclic constraint, gets homogeneous after the potential transformation with respect to a stable strategy. Detecting if there is such a cyclic constraint is easy.

Lemma 7.7 *Let σ and σ' be two strategies such that neither $S(\sigma, \tau)$, nor $S(\sigma', \tau)$ has any cyclic constraints. Assume that σ is stable. Then σ' is no better than σ.*

Proof. Since $S(\sigma', \tau)$ has no cyclic constraints, the dependence of the variables on each other can be described by a DAG G, where the sinks are the variables with constraints of the form $v \leq w$, for a constant w, in $S(\sigma', \tau)$. For each variable v, let $q(v)$ and $q'(v)$ be the value of v under σ and σ', respectively. Let the sequence T of variables be a topological sorting of G. We prove by induction on T that $q'(v) \leq q(v)$ for every variable v. If v is last in T, then v is a sink in G and has a constraint of the form $v \leq w$, for constant w, in $S(\sigma', \tau)$. Since σ is stable we have $w = q'(v) \leq q(v)$.

Assume $q'(u) \leq q(u)$ for all variables u after v in T. Let the constraint for v in $S(\sigma', \tau)$ be $v \leq c_1 \cdot x_1 + \ldots c_k \cdot x_k + w$. Then $q'(v) = c_1 \cdot q'(x_1) + \ldots c_k \cdot q'(x_k) + w$. Since σ is stable we have $q(v) \geq c_1 \cdot q(x_1) + \cdots + c_k \cdot p(x_k) + w$. By inductive assumption, this is no less than $c_1 \cdot q'(x_1) + \cdots + c_k \cdot q'(x_k) + w = q'(v)$. □

Proof of Theorem 7.1. We verify that a system S does *not* have a solution with value at least k as follows: 1) guess an optimal strategy pair (σ, τ); 2) check that σ is stable; 3) check that $S(\tau)$ has a finite optimal solution, and hence that σ is optimal.

If these steps can all be computed in polynomial time (as we actually show below), then the problem is in coNP. Verifying that a strategy is stable is easy, so the question is whether we can determine if a system with only one constraint per uncontrolled variable allows MAX to create unbounded values.

We first check whether there is a strategy σ' such that $S(\sigma', \tau)$ has a cyclic constraint over variables $x_1, \ldots, x_k$ such that in some constraint $x_i \leq \mathit{poly}(\overline{x}, x_{i+1})$, the coefficient in front of x_{i+1} is larger than 1. This can be done in polynomial time by search on a graph representation of $S(\tau)$.

Then we check whether there is a strategy σ', such that $S(\sigma', \tau)$, has a multiple cyclic constraint over some variable. This can be done in the following way. Consider the system $S(\tau)$. For each pair (y, z) of variables appearing together in some constraint, say one for x, use graph reachability to check whether x can be reached from both y and z in the graph representation of $S(\tau)$. The running time for this step is polynomial.

Once we know that there are no cyclic constraints with coefficients larger than 1 and no multiple cyclic constraints over any variable, it remains to check whether there is some single cyclic constraint with only unit coefficients that allows MAX to get an unbounded value. After the potential transformation with respect to σ, we search for a cyclic constraint with only homogeneous constraints. If it exists, the system is unbounded by Lemma 7.6, and the strategy τ we guessed was incorrect. If, on the other hand, there is no such cyclic constraint, then we claim that σ is optimal. Indeed, if there are no cyclic constraints at

all, the claim follows from Lemma 7.7. Otherwise, suppose there is a cyclic constraint, say over $x_1, x_2, \ldots, x_k$, and involving constraints $c_1, \ldots, c_k$. If the cyclic constraint gives unboundedness, it must be possible to improve the values of the variables that belong to the constraints $c_1, \ldots, c_k$, *except* $\{x_1, \ldots, x_k\}$. Recall that such variables cannot depend on any variable in $\{x_1, \ldots, x_k\}$; otherwise multiple cyclic constraints are possible. To show that it is impossible to improve the values of these variables, we can remove all constraints involving $x_1, \ldots, x_k$ and variables that depend on them from the system. By applying the argument recursively to this smaller system S', we know that σ restricted to S' is optimal. This means that no strategy which enforces the cyclic constraint over $x_1, \ldots, x_k$ can give a feasible system. For similar reasons, the same must hold for any other cyclic constraint. Now by Lemma 7.5, σ is optimal in S. Thus we have short witnesses for NO-instances, and Theorem 7.1 follows. □

7.2 Rational Univariate Systems

In a *rational univariate* CLPP instance all constraints have the form $x_i \leq c \cdot x_j + w$, where x_i, x_j are variables and $c \in \mathbb{Q}^+, w \in \mathbb{Q}$ are constants.

Theorem 7.8 *Rational univariate CLPP is in NP∩ coNP.*

Proof. As for systems with integer coefficients above, a pair (σ, τ) of strategies can function as witnesses for NO-instances, as long as

1. the optimal value in $S(\tau)$ is sufficiently small, and
2. σ is stable in $S(\tau)$.

We know that if S is indeed a NO-instance, such a pair must exist by Lemma 3.2. We also know from Theorem 6.1 that if σ is stable in $S(\tau)$, then either it is also optimal, or the optimal solution to $S(\tau)$ is unbounded. Thus, after guessing (σ, τ), it is enough to verify that there is no strategy σ' such that $S(\sigma', \tau)$ is unbounded. We show here that this can be done in polynomial time.

If a univariate system $S(\sigma', \tau)$ is unbounded, then it contains a cyclic dependency $x_1 \leq c_1 \cdot x_2 + w_1$, $x_2 \leq c_2 \cdot x_3 + w_2$, …, $x_k \leq c_k \cdot x_1 + w_k$. Furthermore,

1. either $\prod_{i=1}^{k} c_i > 1$,
2. or $\prod_{i=1}^{k} c_i = 1$, and the weights w_i are *appropriate* (discussed below).

Thus we have to check that there is no σ' such that there is a cyclic dependency in $S(\sigma', \tau)$ that satisfies one of these two conditions. We start by showing how to check the first one.

Consider the graph representation of $S(\tau)$, and remove the MIN-attractor of all sinks (all vertices from which MAX cannot avoid reaching a sink). Now add a supersink t, and an edge from every other vertex in the graph to t. Call this graph G. For every edge e in G corresponding to a constraint in $S(\tau)$, if the coefficient in the constraint was c, give

e weight $-\ln(c)$.[12] Edges to t get weight 0. Now apply the Bellman-Ford algorithm to compute the shortest paths to t. The only result we are interested in is whether there are negative weight cycles in G. Suppose there is such a cycle $e_1, e_2, \ldots, e_{k+1} = e_1$. Then we have $-\ln(c_1) - \ln(c_2) - \cdots - \ln(c_k) < 0$, or equivalently $\ln(c_1) + \ln(c_2) + \cdots + \ln(c_k) > 0$. This means that $\ln(\prod_{i=1}^{k} c_i) > 0$ and thus $\prod_{i=1}^{k} c_i > 1$. If the graph has no negative weight loops, on the other hand, there are no cyclic dependencies in $S(\tau)$ that satisfy the first condition. Details about how to achieve the necessary precision for this test are left to Section 7.2.1.

Now suppose that we have already checked for the first condition, and concluded that no such cyclic dependencies exist. We proceed to checking for the second condition. Suppose a cyclic dependency involves variables $x_1, \ldots, x_k$, coefficients $c_1, \ldots, c_k$, and constants $w_1, \ldots, w_k$. This means that variable x_1 is constrained by $x_1 \leq w_1 + c_1(w_2 + c_2(w_3 + \cdots + c_k x_1)\ldots)$. Written differently, this becomes $x_1 \leq w_1 + c_1 w_2 + c_1 c_2 w_3 + \cdots + c_1 \ldots c_k x_1$. Suppose we have $\prod_{i=1}^{k} c_i = 1$. Then this constraint will be satisfied when

$$0 \leq w_1 + c_1 w_2 + c_1 c_2 w_3 + \cdots + c_1 \ldots c_{k-1} w_k.$$

Suppose that in $S(\sigma, \tau)$ the constraints on the cycle are not used for variables $x_{r_1}, x_{r_2}, \ldots, x_{r_j}$, appearing in that order on the cycle, and that under $S(\sigma, \tau)$ they get values $z_{r_1}, z_{r_2}, \ldots, z_{r_j}$, respectively. Since no switches are attractive,

$$\begin{aligned} z_{r_1} &\geq w_{r_1} + c_{r_1} w_{r_1+1} + \cdots + c_{r_1} \cdot \cdots \cdot c_{r_2-1} z_{r_2}, \\ &\cdots \\ z_{r_j} &\geq w_{r_j} + c_{r_j} w_{r_j+1} + \cdots + c_{r_j} \cdot \cdots \cdot c_{r_1-1} z_{r_1}. \end{aligned}$$

Put together, these inequalities give

$$z_{r_1} \geq w_{r_1} + c_{r_1} w_{r_1+1} + \cdots + c_{r_1} \cdot \cdots \cdot c_{r_1+k-1} z_{r_1},$$

or, equivalently, $0 \geq w_{r_1} + c_{r_1} w_{r_1+1} + \cdots + c_{r_1} \cdot \cdots \cdot c_{r_1+k-2} w_{r_1+k-1}$.

Thus, for x_1 to be unbounded in $S(\sigma', \tau)$, we must have

$$0 = w_1 + c_1 w_2 + c_1 c_2 w_3 + \cdots + c_1 \ldots c_{k-1} w_k.$$

The situation for all other variables on the cycle is similar. This means that all constraints on the cycle will have weight 0 after the potential transformation.

The problem is that we may have other cycles, with $\prod_{i=1}^{k} c_i < 1$, such that all constraints have weight 0 after the potential transformation. We thus have to be able to separate these two kinds of cycles. After making the potential transformation, remove all edges that do not have weight 0. Again remove the MIN-attractor of all sinks. Call the resulting graph G. Assign to each edge e in G the weight $-\ln(c)$ where c is the coefficient of the constraint corresponding to e. Now we want to verify that G does not contain any 0-sum loops.

Since we already know that G cannot contain any negative sum loop, this can be done by subtracting a sufficiently small value from all weights and looking for 0-sum loops, which now become negative, by using the Bellman-Ford algorithm. Again, details about precision are left to Section 7.2.1. □

[12] Recall that all coefficients are nonnegative.

It seems difficult to combine and extend the techniques from Sections 7.1 and 7.2 to cover the general case of several variables with rational coefficients in the right-hand sides of constraints. Multiplying by the least common multiple of all denominators does not help, because this may result in variables with non-unit coefficients on the left. Consequently, analyzing whether there is just one or two cyclic dependencies (as in the integer case) is not enough.

7.2.1 Notes on Precision

When using the logarithms of coefficients as
weights in a graph, the problem is that the logarithms are irrational, and have to be rounded. In this section we describe how to overcome this obstacle.

Suppose in a rational univariate CLPP instance $S(\tau)$, W is the least common multiple of the denominators of coefficients. If there is a cyclic dependency involving coefficients $c_1, \ldots, c_k$ such that $c = \Pi_{i=1}^{k} c_i > 1$, then

$$c \geq 1 + \frac{1}{W^n} = \frac{1 + W^n}{W^n}.$$

Using the derivative of the ln function, we can conclude that

$$\ln c \geq \frac{1}{W^n + 1}.$$

When computing the logarithms of the individual coefficients, we can make sure that the maximal rounding error is smaller than $1/(W^n \cdot n^3)$. Notice that the numbers we compute will have a number of digits polynomial in the input size. After assigning weights to the edges in a graph as described above, we can add $1/(W^n \cdot n^2)$ to all weights. If the sum $-\ln c$ of weights on a loop would have been negative, assuming no rounding errors, it will be negative in our construction, since

$$\frac{1}{W^n + 1} > \frac{n}{W^n \cdot n^2} + \frac{n}{W^n \cdot n^3}.$$

Nonnegative loops obviously stay nonnegative.

When checking for 0-sum loops when we know that there are no negative value loops, we can again ensure that the maximal rounding error is smaller than $1/(W^n \cdot n^3)$. Using the derivative of the ln function, we know that if a loop sum is positive, then it is at least $1/W^n$. Thus, if we subtract $1/(W^n \cdot n^2)$ from all weights, only 0-sum loops will become negative.

8 Conclusions

The following problems remain open for the CLPP.

1. Does the problem of determining whether the optimal solution to a given arbitrary CLPP instance is *finite* belong to NP ∩ coNP? For the integer coefficient and rational univariate cases the answer is positive, as well as for all CLPP instances derived from games.

2. Is it possible, by employing "more structure" of the CLPP, to improve on the subexponential analysis? Our current analysis is almost oracle-based, relying only on the facts that: 1) every facet of a CLPP instance is a CLPP instance; 2) all facets are linearly ordered by the value of the best strategy on the facet; 3) every stable strategy is optimal.

3. Interior point schemes should be investigated further, as per their potential advantage over discrete schemes (as in the case of LP). Although we know how to make interior point search subexponential, there are further problems. It is conceivable that the interior point algorithms can make exponentially many discretization steps before finding the optimal solution. We hope this is not the case in the restricted cases of the CLPP.

4. The integer coefficient and rational univariate CLPPs are in NP ∩ coNP. Can it also be solved in expected subexponential time?

5. Whether the CLPP is polynomial time solvable? If it is then so are parity, mean payoff, discounted payoff, simple stochastic games, and *Longest-Shortest Paths*. Can any lower bounds (under strong complexity-theoretic assumptions) be set for the CLPP?

6. The CLPP captures many important optimization problems, including several games. Are there any other interesting controlled optimization problems of this kind?

Acknowledgments We are grateful to Leonid Khachiyan, Vladimir Gurvich, and Endre Boros for inspiring discussions and illuminating ideas. We thank DIMACS for providing an excellent research environment.

References

[1] H. Björklund. *Combinatorial Optimization for Infinite Games on Graphs*. PhD thesis, Uppsala University, 2005.

[2] H. Björklund, O. Nilsson, O. Svensson, and S. Vorobyov. Controlled linear programming: Boundedness and duality. Technical Report DIMACS-2004-56, DIMACS: Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University, NJ, December 2004. `http://dimacs.rutgers.edu/TechnicalReports/`.

[3] H. Björklund, O. Nilsson, O. Svensson, and S. Vorobyov. The controlled linear programming problem. Technical Report DIMACS-2004-41, DIMACS: Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University, NJ, September 2004. `http://dimacs.rutgers.edu/TechnicalReports/`.

[4] H. Björklund and S. Sandberg. Algorithms for combinatorial optimization and games adapted from linear programming. In B. ten Cate, editor, *Proceedings of the Eighth European Summer School on Logic Language, and Information (ESSLLI'2003) Student Session*, 2003.

[5] H. Björklund, S. Sandberg, and S. Vorobyov. Optimization on completely unimodal hypercubes. Technical Report 2002-018, Uppsala University / Information Technology, May 2002. `http://www.it.uu.se/research/reports/`.

[6] H. Björklund, S. Sandberg, and S. Vorobyov. Complexity of model checking by iterative improvement: the pseudo-Boolean framework. In M. Broy and A. Zamulin, editors, *Andrei Ershov Fifth International Conference "Perspectives of System Informatics"*, volume 2890 of *Lecture Notes in Computer Science*, pages 381–394, 2003.

[7] H. Björklund, S. Sandberg, and S. Vorobyov. A discrete subexponential algorithm for parity games. In H. Alt and M. Habib, editors, *20th International Symposium on Theoretical Aspects of Computer Science, STACS'2003*, volume 2607 of *Lecture Notes in Computer Science*, pages 663–674, Berlin, 2003. Springer-Verlag. Full preliminary version: TR-2002-026, Department of Information Technology, Uppsala University, September 2002.

[8] H. Björklund, S. Sandberg, and S. Vorobyov. On combinatorial structure and algorithms for parity games. Technical Report 2003-002, Department of Information Technology, Uppsala University, 2003. `http://www.it.uu.se/research/reports/`.

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

[10] H. Björklund, S. Sandberg, and S. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. In J. Fiala, V. Koubek, and J. Kratochvil, editors, *29th International Symposium on Mathematical Foundations of Computer Science (MFCS'2004)*, volume 3153 of *Lecture Notes in Computer Science*, pages 673–685. Springer, August 2004. Full version to appear in *Discrete Applied Mathematics*. Preliminary full version: TR DIMACS-2004-05 `http://dimacs.rutgers.edu/TechnicalReports/`.

[11] H. Björklund, S. Sandberg, and S. Vorobyov. Randomized subexponential algorithms for infinite games. Technical Report DIMACS-2004-09, DIMACS: Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University, NJ, April 2004. `http://dimacs.rutgers.edu/TechnicalReports/`. To appear in *Theoretical Computer Science*.

[12] H. Björklund, O. Svensson, and S. Vorobyov. Linear complementarity algorithms for mean payoff games. Technical Report DIMACS-2005-05, DIMACS: Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University, NJ, February 2005. `http://dimacs.rutgers.edu/TechnicalReports/`.

[13] A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.

[14] A. Condon. On algorithms for simple stochastic games. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 13:51–71, 1993.

[15] R. W. Cottle, J.-H. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Academic Press, 1992.

[16] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journ. of Game Theory*, 8:109–113, 1979.

[17] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1996.

[18] V. A. Gurvich, A. V. Karzanov, and L. G. Khachiyan. Cyclic games and an algorithm to find minimax cycle means in directed graphs. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 28(5):85–91, 1988.

[19] A. J. Hoffman and R. M. Karp. On nonterminating stochastic games. *Management Science*, 12(5):359–370, 1966.

[20] G. Kalai. A subexponential randomized simplex algorithm. In *24th ACM STOC*, pages 475–482, 1992.

[21] Feng-Tien Yu Katta G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, Berlin, 1988. http://ioe.engin.umich.edu/people/fac/books/murty/linear_complementarity_webbook/.

[22] W. Ludwig. A subexponential randomized algorithm for the simple stochastic game problem. *Information and Computation*, 117:151–155, 1995.

[23] J. Matoušek, M. Sharir, and M. Welzl. A subexponential bound for linear programming. In *8th ACM Symp. on Computational Geometry*, pages 1–8, 1992.

[24] V. Petersson and S. Vorobyov. Interior-point approach to parity games. In *IEEE Annual Symposium on Logic in Computer Science (LICS'2001)*, Boston, Massachusetts, June 2001. Short communication.

[25] V. Petersson and S. Vorobyov. Parity games: interior-point approach. Technical Report 008, Uppsala University / Information Technology, May 2001. `http://www.its.uu.se/research/reports/`.

[26] N. Pisaruk. Mean cost cyclical games. *Mathematics of Operations Research*, 24(4):817–828, 1999.

[27] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, 1986.

[28] L. S. Shapley. Stochastic games. *Proc. Nat. Acad. Sci. USA*, 39:1095–110, 1953.

[29] U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158:343–359, 1996.