DIMACS Technical Report 2004-41
September 2004

The Controlled Linear Programming Problem

by

Henrik Björklund, Olle Nilsson, Ola Svensson, Sergei Vorobyov [1,2,3,4,5]

[1]DIMACS visitors, August, 2004.
[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*" and "*Interior-Point Methods for Infinite Games*".
[4]DIMACS hosts: Leonid Khachiyan and Endre Boros
[5]Email: `vorobyov@csd.uu.se`

ABSTRACT

We introduce and investigate the new Controlled Linear Programming Problem. In a system of linear constraints of the form $x_i \leq p_i^j(\bar{x}) + w_i^j$, where p_i^j are linear homogeneous polynomials with nonnegative coefficients, some variables are *controlled* and the controller wants to select exactly one constraint for each controlled variable in a way that makes $\max \sum x_i$ subject to the remaining constraints as large as possible (over all selections). We suggest several iterative strategy improvement policies (simplex-like and interior-point), prove optimality conditions in several important cases, describe subexponential algorithms, and interior point ones, and show that the decision version of the problem is a member of NP$\cap$coNP whenever stability yields optimality, and also when all coefficients are nonnegative integers.

It turns out that the well known mean payoff, discounted payoff, and simple stochastic games are easily reducible to the Controlled LP-Problem. This suggests a simple unifying view of all these games as particular restricted instances of the problem.

We show that a slight generalization of the problem allowing for negative coefficients in the constraint polynomials p_i^j leads to NP-hardness.

Contents

1 Introduction

Consider a succinctly represented family $\mathcal{F} = \{P_i\}_{i \in I}$ of instances of a polynomially solvable optimization problem. 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 P_i's be weighted digraphs with distinguished source s and sink t. Find the graph with the *longest* shortest s-t-distance. As an example of succinct representation, consider a digraph G with a distinguished subset of *controlled* vertices C, and generate P_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, called the *Longest Shortest Paths (LSP)* problem, has useful applications and is investigated in [4].

The example of a *controlled* problem above, in which the controller selects options in order to maximize 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. Natural and simple as these problems sound, they turn out to be NP-hard; see Section 12.

The questions which immediately appear are as follows. 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 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 in the LSP problem [5]. 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 solutions in the case of the LSP problem.

Now the main question. Are there any other useful *controlled* optimization problems with the positive (satisfactory) answers to the questions above? Do they have good and interesting applications?

In this paper we introduce and investigate another problem of this kind, dubbed the Controlled LP-Problem. 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 *controlled* and the controller wants to select exactly one constraint for each controlled variable in a way that makes $\max \sum x_i$, subject to the remaining constraints, as large as possible (over all selections).

It turns out that the well known mean payoff, discounted payoff, and simple stochastic games [9, 11, 7, 22], as well as the LSP problem [5], are easily reducible to the Controlled LP-Problem. This suggests a simplified unifying view of all these games as particular restricted instances of the problem, and proves its usefulness. The subexponential algorithms for the CLPP (Section 11) are immediately applicable to solve all the above games.

The CLPP with nonnegative coefficients demonstrates a robust structure amenable to a

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

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 3, 4). As a result, the problem of finding local maxima for the CLPP in this case can be solved in subexponential time (Section 11). In many interesting cases of the CLPP with nonnegative coefficients, we also prove that a stable (not locally improvable) pure strategy (selection) of constraints is also globally optimal (Section 7). In these cases we can find global maxima in subexponential time, and the corresponding decision problem (is there a selection providing for the optimum value above the threshold) is in NP∩coNP (Section 7.4). We also investigate a case of the CLPP with nonnegative integer coefficients, which is in NP∩coNP, but without the "stability ⇒ optimality" property and without known subexponential iterative improvement algorithms (Section 8). This shows, in particular, that there is a CLPP subclass, which belongs to NP∩coNP, while being more general than any of the games mentioned above.

The next natural step in generalization leads to (presumable) intractability. We show 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 NP∩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-Partitite Matching and Controlled Max-Flow confirm this classification (Section 12).

Notations, Conventions. By boldface $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{1}$ we denote column vectors of variables x_i, y_i, and ones, and by $\mathbf{x}^T$ the transposed row vector. A linear polynomial is called *homogeneous* if it contains no free members, i.e., of the form $\mathbf{c}^T\mathbf{x}$.

2 The Controlled Linear Programming Problem

Let $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_m)$ be two vectors of variables, called *controlled* and *uncontrolled*, respectively. Let $p_i^j(\mathbf{y})$ and $q_k^l(\mathbf{x})$ be homogeneous linear polynomials with *non-negative* real coefficients, and $w_i^j, w_k^{\prime l} \in \mathbb{R}$, for $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, n_i\}$, $k \in \{1, \ldots, m\}$, $l \in \{1, \ldots, m_k\}$. A linear constraint is called:

- *optional* if it has form $x_i \leq p_i^j(\mathbf{y}) + w_i^j$, and
- *compulsory* if it has form $y_k \leq q_k^l(\mathbf{x}) + w_k^{\prime l}$.

(For notational convenience we assume "bi-partiteness": x_i's and y_j's alternate in constraint sides. This is no loss of generality and can always be achieved by introducing new variables.)

Definition 2.1 (Controlled LP-Problem) *Given the data as above, find a tuple, called a* positional strategy*, $(j_1, \ldots, j_n) \in \prod_{j=1}^{n}\{1, \ldots, n_j\}$ that renders maximal (over all possible such tuples) the value of the objective function $\sum_{i=1}^{n} x_i + \sum_{i=1}^{m} y_i$ subject to the system of constraints $S(j_1, \ldots, j_n)$ consisting of all compulsory and the optional constraints $x_i \leq p_i^{j_i}(\mathbf{y}) + w_i^{j_i}$ for $i = 1, \ldots, n$.* □

Quite often we want to maximize the values of all variables. This is not the same as maximizing their sum in case some variable is unbounded. Section 3.1 explains how to proceed in this case.

Proviso. To avoid immediate unboundedness of a feasible constraint system, we assume that every variable occurs in the left-hand side of some constraint. □

The name for this problem comes from the intuition that the controller selects *exactly one* optional constraint for each controlled variable x_i to maximize the objective function. In general, some selections of optional constraints may lead to infeasible systems (the value of objective function is $-\infty$), some to polyhedra unbounded in some coordinate directions (the value $+\infty$), and some to nonempty polyhedra with a finite optimum of the target function.

It would be more appropriate (but would overload the terminology) to call the problem the Positive Controlled LP-Problem, since we impose the positiveness of the coefficients in polynomials. We show that this restriction is crucial for efficiency; see Section 9. Whenever it causes no confusion, we omit the adjective "positive".

Definition 2.1 introduces the optimization problem. As usual, the corresponding decision version is defined by giving an additional bound and asking whether there exists a selection rendering the objective function larger than that bound.

3 Discrete Switching Algorithms

A selection $\lambda = (j_1, \ldots, j_n)$ of optional constraints, is called *a pure*[2] *strategy.* Denote $S(j_1, \ldots, j_n)$ the resulting system of constraints obtained by taking all compulsory constraints and leaving only the j_k-th optional constraint for each variable x_k, $k = 1, \ldots, n$. Say that a strategy λ is *admissible* if $S(\lambda)$ is feasible (has a solution).

Consider an optimal solution $(\mathbf{x}^*, \mathbf{y}^*)$ to a feasible system $S(j_1, \ldots, j_n)$, maximizing the target function $\mathbf{1}^T\mathbf{x} + \mathbf{1}^T\mathbf{y}$, symbolically,

$$(\mathbf{x}^*, \mathbf{y}^*) = \arg\max\{\mathbf{1}^T\mathbf{x} + \mathbf{1}^T\mathbf{y} \text{ subject to } S(j_1, \ldots, j_n)\}.$$

Say that a *switch* for variable x_k from j_k-th to j'_k-th constraint is *attractive*, if

$$p_k^{j_k}(\mathbf{y}^*) + w_k^{j_k} < p_k^{j'_k}(\mathbf{y}^*) + w_k^{j'_k}. \tag{1}$$

Say that a sequence of switches for variables $x_1, \ldots, x_n$ from $(j_1, \ldots, j_n)$ to $(j'_1, \ldots, j'_n)$ is *profitable*, if for each variable v in $\mathbf{x}$, $\mathbf{y}$ one has

$$\max\{v \text{ subject to } S(j_1, \ldots, j_n)\} \quad \leq \quad \max\{v \text{ subject to } S(j'_1, \ldots, j'_n)\}, \tag{2}$$

and for some variable the inequality (2) is *strict.*

We have the following two important theorems, underlying a wide range of iterative improvement algorithms for the controlled LP-problem. The first theorem guarantees strict monotonic improvement of attractive switches. It will be the corollary of the more general theorem in Sections 5.

Theorem 3.1 *Every sequence of attractive switches is profitable and transforms an admissible strategy into an admissible one.* □

Say that an admissible strategy is *stable* if it does not have attractive switches. For many important cases of the Controlled LP-Problem, discussed below, we have the following polynomial-time optimality test.

Theorem 3.2 *Every stable strategy is optimal.* □

Theorems 3.1 and 3.2 immediately suggest the following correct and complete generic algorithm for solving, in some cases, the Controlled LP-Problem.

Generic Algorithm for Controlled LP

1. Start with any admissible initial strategy (we discuss below how to select such strategies.)

[2]or discrete

2. Proceed by making any attractive switches, until obtaining a stable strategy. (Simplify the system if necessary by getting rid of unbounded variables and associated constraints. It is essential that this simplification takes place only after reaching a stable strategy; see Section 3.1.)

3. In many important cases (Sections 7) the resulting stable strategy is also *optimal.*

Specific iterative strategy improvement algorithms are obtained by imposing a policy on selecting attractive switches at every iteration. Some possible policies are: single random switch, all attractive switches, random multiple attractive switches (Section 11.1). We also discuss two randomized subexponential algorithms for the Controlled LP-Problem, based on adapting the schemes for LP due to Kalai [13] and Matoušek, Sharir, Welzl [16] (Section 11.2). In general, there are examples of "poor" switching policies leading to an exponential number of switches [5], so the problem is by no means trivial.

3.1 Eliminating Unbounded Variables and Constraints

In many circumstances we want to maximize the values of all variables, rather than just the sum of all variables. In this case we need to get rid of the unbounded variables, simplify the system, and proceed with the remaining variables. Here is the recipe.

Theorem 3.3 *Eliminating unbounded variables and constraints from the system $S(\lambda)$, does not affect the optimal values of the remaining variables, provided that λ is stable.*

Proof. Assume the value of a controlled variable x_i is bounded in $S(\lambda)$. 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 $S(\lambda)$. Then it has a bounded constraint. Since the unbounded constraints never will tighten the value of y_i, even if the bounded constraints go unbounded, we can safely remove them. $\Box$

Eliminating unbounded variables allows us to assume in the premises of theorems below, without loss of generality, that a system has a finite optimal solution under a current strategy.

It is easy to construct an example where an elimination in a system with an unstable strategy has a devastating effect to the solution. Consider the unstable strategy ((1,0),(1,0), (0,1)) in Figure 1 in Section 6 and the notational explanations there. Eliminating the unbounded variables and constraints will give v_3 the final value of 1, while its correct value is infinity.

3.2 Retreats: Boundedness and Feasibility

An alternative way of avoiding unboundedness is to add *retreat* constraints for the uncontrolled variables. They are called retreats because we can imagine a minimizing player (opponent) choosing constraints for the uncontrolled variables, and using the retreats only when the value for a variable threatens to become unbounded.

To do this, compute an upper bound M on the largest *finite* value a variable can take in the solution to the linear program arising from any strategy of the maximizer. Then add a constraint $y_k \leq M$, for each uncontrolled variable y_k. After solving the new system, any variable that gets a value close to M has an unbounded value in the optimal solution to the original problem.

Similarly, we can add retreats for the maximizer, in order to ensure that there is at least one strategy that yields a feasible linear program. Add a constraint $x_i \leq -M$ for each controlled variable x_i. Now the strategy that for every controlled variable selects the retreat constraint is *admissible*.

4 Interior Point Algorithms

The previous discrete switching algorithms can be seen as a simplex-like, visiting the vertices of the Cartesian product $\prod_{i=1}^{n}\{1,\ldots,n_i\}$ (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 by us in the context of simple stochastic games in [19, 18]).

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:

$$\sum_{j=1}^{n_i} \lambda_i^j (p_i^j(\mathbf{y}) + w_i^j). \tag{3}$$

Collect all the constraints (3), 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 $\bar{\lambda}_i$ with 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 three facts, guaranteeing the correctness of a wide range of interior point algorithms:

1. for an admissible strategy, any combination of attractive switches leads to an admissible strategy and a better target function value (Section 5);

2. for any stable mixed strategy (no attractive switches) there is a pure no-worse stable strategy (Section 6);

3. every pure stable strategy is optimal, in many important cases discussed in Section 7.

Now the Generic Algorithm from Section 3 works perfectly well for the generalized case of interior-point switches.

5 Attractiveness and Profitability

Definition 5.1 *Given an admissible interior point strategy* $\lambda = (\bar\lambda_1, \bar\lambda_2, ..., \bar\lambda_n)$*, let* $(\mathbf{x}^*, \mathbf{y}^*)$ *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*

$$\sum_j \lambda_i^j (p_i^j(\mathbf{y}^*) + w_i^j) < \sum_j \lambda_i'^j (p_i^j(\mathbf{y}^*) + w_i^j). \tag{4}$$

Say that the switch from λ *to another strategy* $\lambda' = (\bar\lambda'_1, \ldots, \bar\lambda'_n)$ *is* profitable*, if* λ' *is admissible and for each variable* v *in* $\mathbf{x}$*,* $\mathbf{y}$ *one has*

$$\max\{v \textit{ subject to } S(\lambda)\} \quad \le \quad \max\{v \textit{ subject to } S(\lambda')\}, \tag{5}$$

and for some variable the inequality (5) *is* strict*.* □

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

Proof. Assume attractive switches were made, from strategy λ to λ'. Let $(\mathbf{x}^*, \mathbf{y}^*)$ be the optimal solution to the old LP $S(\lambda)$. For every switched variable x_i, we have the old constraint $x_i \le \sum_j \lambda_i^j (p_i^j(\mathbf{y}) + w_i^j)$, and the new constraint $x_i \le \sum_j \lambda_i'^j (p_i^j(\mathbf{y}) + w_i^j)$. By definition of an attractive switch, we have (4) satisfied.

Then it is easy to see that $(\mathbf{x}^*, \mathbf{y}^*)$ 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 (4).

Take one switched variable x_i, let $x'_i = x_i + \varepsilon$, for a small enough $\varepsilon > 0$. Then $(\mathbf{x}^*, \mathbf{y}^*)$, where x_i in $\mathbf{x}^*$ is replaced by x'_i 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 $p_i^j(\mathbf{y})$ and $q_k^l(\mathbf{x})$ 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 and q_i^j in right-hand sides of constraints have nonnegative coefficients. The above argument fails, if negative coefficients are allowed. □

Now Theorem 3.1 follows immediately, as a corollary.

6 Discretization

In this section we show that every stable mixed (interior-point) strategy λ can be made discrete (pure). This simplification allows us to concentrate on discrete/pure strategies while proving optimality of stable strategies in Section 7.

Theorem 6.1 *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 $(\mathbf{x}^*, \mathbf{y}^*)$. 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 $p_i^j(\mathbf{y}^*) + w_i^j$ 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 $x_i \leq \sum_{j=1}^{n_i} \lambda_i^j (p_i^j(\mathbf{y}^*) + w_i^j) = \sum_{j=1}^{n_i} \lambda_i^j z = 1 \cdot z = p_i^k(\mathbf{y}^*) + w_i^k$, switching from $\bar{\lambda}_i$ to $\bar{\lambda}'_i$ leaves the solution $(\mathbf{x}^*, \mathbf{y}^*)$ 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 Figure 1 we give the graph representation of the following controlled linear program, where all constraints are optional:

$$\begin{cases} v_1 & \leq & v_2, \\ v_1 & \leq & 0, \end{cases} \qquad \begin{cases} v_2 & \leq & v_1, \\ v_2 & \leq & 0, \end{cases} \qquad \begin{cases} v_3 & \leq & v_2, \\ v_3 & \leq & 1. \end{cases}$$

As an example, a mixed (actually, pure) strategy $((1,0),(1,0),(0,1))$ selects the first constraints for $v_{1,2}$ and the second for v_3.

It is conceivable, as Figure 1 shows, that after discretization described in the proof above the resulting strategy becomes *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.

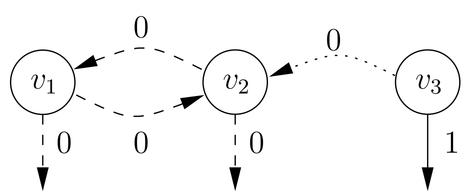

Figure 1: In this controlled LP there is a strategy that is stable before but not after discretization. Round vertices are controlled.

Consider the mixed strategy $((1/2,1/2),(1/2,1/2),(0,1))$, in which the controller selects for $v_{1,2}$ each of the optional constraints with equal probability 1/2, and for v_3 purely selects $v_3 \leq 1$. This strategy is stable yielding the optimal target value 1, but its discretization to $((1,0),(1,0),(0,1))$ becomes unstable, the attractive switch in v_3 to $v_3 \leq v_2$ makes all three variables unbounded.

7 Stability and Optimality

As discussed in Section 7.5, in general, a stable strategy in an instance of the Positive Controlled LP-Problem is *not necessarily optimal*. In this and subsequent sections we also present several important particular cases when the desired implication "stability $\Rightarrow$ optimality" holds, i.e., a stable strategy is guaranteed to be optimal. This suggests an easy polynomial-time verification of optimality, used by all our algorithms in Section 11.

7.1 Univariate Discounted Case

Let $0 < \delta < 1$ be fixed, and consider the Positive Controlled LP-Problem with restricted right-hand sides of one of the forms:

$$x \leq w + \delta \cdot y, \tag{6}$$

$$y \leq w + \delta \cdot x, \tag{7}$$

$$y \leq 0, \tag{8}$$

where x is a controlled variable, y is a noncontrolled variable, and $w \in \mathbb{R}$. In other words, we restrict the number of variables in the right hand sides to at most one, with the coefficient smaller than one. Note that constraints of the form $y \leq w$ may be introduced as abbreviations for $y \leq w + \delta x'$, $x' \leq 0 + \delta y'$, $y' \leq 0$.

Theorem 7.1 *Every discrete/pure stable strategy of the controller (*Max*) in the* Positive Controlled LP-Problem *instance with constraints of the form* (6) – (8) *is optimal.*

Proof. Let λ be a stable strategy in a problem instance S. Assume $S(\lambda)$ is feasible, and let $(\mathbf{x}^*, \mathbf{y}^*)$ be an optimal solution. We claim and prove shortly that this optimal solution is *finite*.

A function p mapping variables in $\mathbf{x}$ and $\mathbf{y}$ to $\mathbb{R}$ is called a *potential function* for the system $S(\lambda)$ if for every constraint $u \leq w + \delta \cdot v$ ($y \leq 0$, resp.) one has $p(u) \leq w + \delta \cdot p(v)$ ($p(y) \leq 0$, resp.).

It is easy to see that the components of a finite optimal solution to $S(\lambda)$ determine potentials for the system $S(\lambda)$. Let us take the potential function p obtained this way and make the following *potential transformation* to the whole system S:

- if $u \leq w + \delta \cdot v$ is a constraint, change it to $u \leq w' + \delta \cdot v$, where $w' = w - p(u) + \delta \cdot p(v)$.

Since the strategy λ is stable, after the potential transformation, for every constraint $x \leq w' + \delta \cdot y$ that is *not* in λ one has $w' \leq 0$, with $w = 0$ for constraints in λ. This follows from the fact that stability means choosing a constraint with the maximal right-hand side (no attractive switches). Define an *optimal counterstrategy* $\tau(\lambda)$ of the controller's opponent against λ as a selection, for each variable, of one of the compulsory constraints that are *tight* in the optimal solution for $S(\lambda)$ (i.e., satisfied as equalities). This means that in a strategy $\tau(\lambda)$ in system $S(\lambda)$ there are also *no attractive switches* for the opponent (defined symmetrically as for the controller). It follows that after the potential transformation for each constraint of the controller's opponent $y \leq w' + \delta \cdot x$ one has $w' \geq 0$, with $w' = 0$ for constraints in $\tau(\lambda)$.

Suppose the controller and his opponent, starting in a variable v_0, play a version of Generalized Geography constructing a (finite or infinite) sequence of variables $v_0, v_1, v_2, \ldots$ as follows. The owner of v_i selects a constraint $v_i \leq w_{i,i+1} + \delta \cdot v_{i+1}$; this determines the next variable in the sequence, or the sequence terminates, if the constraint $v_i \leq 0$ is chosen.

The cost $c(v_0, v_1, \ldots)$ of each such finite or infinite sequence is determined as the sum $\sum_i \delta^i w_{i,i+1}$. Note immediately (by telescoping) that the potential transformation $p(.)$ changes the costs to:

1. $c'(v_0, \ldots, v_{l+1}) = c(v_0, \ldots, v_{l+1}) - p(v_0) + \delta^l \cdot p(v_{l+1})$, for a finite sequence, and
2. $c'(v_0, v_1 \ldots) = c(v_0, v_1, \ldots) - p(v_0)$, for an infinite one.

Therefore, the costs of paths after a potential transformation of S shift by a constant, in particular, when the controller and the opponent follow the pair of positional strategies λ and $\tau(\lambda)$ the cost of the path π they form equals zero. Therefore, the cost of the same path before the transformation is $p(v_0) - p(v_{l+1})$ or $p(v_0)$, depending on whether it is finite or infinite.

Suppose now, toward the contradiction that the strategy λ is not optimal, and there is a better strategy λ' providing for a larger value of the target function, consequently, for some variable v_0. Let the controller use this strategy, but the opponent sticks to the previous strategy $\tau(\lambda)$. Then the cost of the path π' they construct in the transformed S consists of the zero-weight edges consistently chosen by the opponent plus nonpositive edge weights chosen according to λ'. Therefore, the cost of this path π' is nonpositive, i.e., no better than of π. It follows, that the same relation holds for the cost of paths π and π' in the system S before the transformation, i.e., π' is no better for the controller. Therefore, λ' is no better than λ. This contradiction shows that λ is optimal.

It remains to show that for any feasible $S(\lambda)$ its optimal solution is finite. Indeed, no matter which constraints the parties select, the cost of a finite or infinite path followed is bounded by $W \cdot \sum_i \delta^i \leq W/(1-\delta)$, where W is the biggest constant w occurring in the constraints S. □

7.2 Multivariate Discounted Case

Consider the *multivariate discounted case*, when constraints are allowed to have form either $u \leq c$ (c a constant), or:

$$u \leq \sum_i a_i \cdot v_i, \tag{9}$$

where u and v_i's are variables, and a_i's are positive constants such that $\sum_i a_i < 1$.

Note that we can equivalently allow for only one type of constraints, namely:

$$u \;\; \leq \;\; \sum_i a_i \cdot v_i + w, \tag{10}$$

because such a constraint is expressible via $u \leq \sum a_i v_i + a'v'$ and $v' \leq w/a'$, where $a' > 0$ is small enough.

Theorem 7.2 *Let S be an instance of the multivariate discounted* Positive Controlled LP-Problem*. Then every discrete/pure stable strategy λ is optimal.*

Proof. Suppose λ is a stable discrete strategy, the system $S(\lambda)$ is feasible, and let $(\mathbf{x}^*, \mathbf{y}^*)$ be an optimal solution to $S(\lambda)$. We will shortly prove that this optimal solution is *finite*.

Generalizing the previous definition, a function p mapping variables in $\mathbf{x}$ and $\mathbf{y}$ to $\mathbb{R}$ is called a *potential function* for the system $S(\lambda)$ if for every constraint $u \leq \sum_i a_i v_i$ ($y \leq c$, resp.) one has $p(u) \leq \sum_i a_i p(v)$ ($p(y) \leq c$, resp.).

It is easy to see that the components of a finite optimal solution to $S(\lambda)$ determine potentials for the system $S(\lambda)$. Let us take the potential function p obtained this way and make the *potential transformation* defined below to the whole system S (rather than $S(\lambda)$).

To define an appropriate potential transformation we first need to introduce auxiliary terminology. For our purposes, a directed *hypergraph* consists of vertices (one per variable in S) and directed hyperedges of the form $(u, a_1v_1, \ldots, a_kv_k, w)$ where u is the source, $v_1, \ldots, v_k$ are weighted targets, and $w \in \mathbb{R}$ is the hyperedge weight. A strategy for the controller (and the opponent) is a selection of a exactly one hyperedge corresponding to the associated constraint.

Now the desired potential transformation is defined by changing the hyperedge $(u, a_1v_1, \ldots,$ $a_kv_k, w)$ weights as follows

$$w' \;\; = w - p(u) + \textstyle\sum_i a_i p(v_i) \tag{11}$$

(Note that $y \leq c$ also forms a hyperedge with source y, no targets, of weight c, which is transformed to $c' = c - p(y)$.)

Let hyperedge $(u, a_1v_1, \ldots,\; a_kv_k, w)$ be selected in λ, and $(u, a_1'v_1', \ldots,\; a_k'v_k', \bar{w})$ be any other hyperedge from u. Then, by definition of $p(.)$ and since λ is stable, we have

$$p(u) = \sum_i a_i p(v_i) + w \geq \sum_i a_i' p(v_i') + \bar{w}.$$

As a consequence, all hyperedges in λ get weight exactly 0, and all other controller hyperedges get nonpositive weights.

Define an optimal counterstrategy $\tau(\lambda)$ against λ for the opponent as a selection, for every variable, of a hyperedge corresponding to any of his tight constraints (holding as equalities) in the optimal solution $(\mathbf{x}^*, \mathbf{y}^*)$ for $S(\lambda)$. For each such hyperedge $(u, a_1v_1, \ldots,\; a_kv_k, w)$ and any other hyperedge λ, and $(u, a_1'v_1', \ldots,\; a_k'v_k', \bar{w})$ from u, we have

$$p(u) = \sum_i a_i p(v_i) + w \leq \sum_i a_i' p(v_i') + \bar{w}.$$

Therefore, the potential transformation makes weights of hyperedges in $\tau(\lambda)$ exactly zero, and all other opponent hyperedges get nonnegative weights.

Now suppose, toward a contradiction, that λ is not optimal, and there is a better strategy λ' for the controller, providing for a larger value of the target function, hence, a larger value for some variable v_0.

When the controller and his opponent, starting from v play a TREE-LIKE GENERALIZED GEOGRAPHY, then the cost of such a play is defined as a sum of weights of all hyperedges traversed. In particular, when they follow the pair of strategies λ, $\tau(\lambda)$, the total sum of all hyperedges is zero, because only 0-weight hyperedges are used by both parties. Suppose, however, that the controller deviates from λ to the allegedly better λ', but the opponent consistently sticks to $\tau(\lambda)$. Then the cost of the game cannot become positive, since the controller does not have positive-weight hyperedges, and the opponent only plays 0-weight hyperedges. Therefore, λ' cannot provide an improvement (larger value) compared to what λ does. This contradiction shows that λ is optimal. To complete the proof it remains to notice that:

1. The cost of the same play π in a potentially transformed and non-transformed systems differ only by a constant (by telescoping).

2. The cost of any play is bounded.

These facts follow from the assumption that $\max \sum_i a_i = \delta < 1$ and therefore the total cost of a play cannot exceed $C \cdot \sum_i \delta^i = C/(1-\delta)$. □

7.3 Univariate Non-Discounted 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:

$$u \;\leq\; w + v, \tag{12}$$

where u, v are variables, and $w \in \mathbb{Z}$. Besides, we allow one (or more) compulsory constraint of the form

$$t \;\leq\; 0. \tag{13}$$

Due to our proviso that every variable appears in the left-hand-side of some constraint, the variables in (13) are the only possible sinks in a directed graph built on variables with directed edges corresponding to inequalities.

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

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

Imposing this condition does not lead to a loss of generality. Indeed, suppose there are zero-weight loops in a graph or, we do not know[3]. We will describe the edge weight transformation $\Delta^{+1}_{\times n}$ that transforms all zero-weight loops into positive-weight ones[4], and preserves the signs of weights of all other loops (positive are turned into positive and negative to negative). The desired transformation is as follows. Multiply all edge weights by n and add one, where n is the number of vertices (variables). Every negative weight loop in the original graph remains negative, because its weight is the transformed graph is $\leq -1 \cdot n + 1 \cdot (n-1) = -1$. All non-negative loops obviously become positive. Now, if an optimal loop in the system before the transformation has weight 0 (the target function has the $+\infty$ value in this case) then the same loop will be "slightly" positive and optimal in the $\Delta^{+1}_{\times n}$-transformed system, and also results in an unbounded target value.

Theorem 7.3 *Let λ be a discrete stable strategy, the graph determined by S does not have zero-weight cycles, $S(\lambda)$ is feasible and has a finite[5] optimal solution. Then λ is optimal.*

Proof. Suppose, λ is a stable discrete strategy. Consider the potential p defined on variables by giving them the respective values of an optimal solution. For every edge/constraint $u \leq w + v$ the potentials satisfy $p(u) \leq w + p(v)$, with equalities for the tight constraints (e.g., for those selected in λ.) Define an optimal counterstrategy $\tau(\lambda)$ for the opponent as any selection of exactly one tight constraint per his variable. Make a potential transformation of the graph of S, by changing the weight of each edge according to $w'_{u,v} = w_{uv} - p(u) + p(v)$. After this transformation: 1) all edges in λ become 0-weight, and all other controller edges become non-positive (by stability); 2) all edges in $\tau(\lambda)$ become 0-weighted, and all other edges of the opponent become non-negative; 3) weights of all loops remain *unchanged*; 4) weights of finite paths $v_0, \ldots, v_l$ change by a constant $-p(v_0) + p(v_l)$ (telescoping), hence the relation $(<, =, >)$ between costs of two paths to a sink remains unchanged.

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

1. either *finite*, terminating in a sink (13); in this case the value for v_0 is non-positive, and this contradicts to the assumption that λ' provides a better value for v_0 than λ;

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

[3]Note that our definition of a zero-weight loop assumes that the controller and the opponent cooperate in constructing such loops, which they actually may not want.

[4]For other applications in Section 10.3 we analogously turn, by $\Delta^{-1}_{\times n}$ (which subtracts rather than adds 1), zero-weight loops to negative-weighted and preserve weight signs of all other loops.

[5]This assumption is no loss of generality; see the end of Section 3.1.

during the potential transformation; since only non-positive edges are used by λ' and $\tau(\lambda)$, the only possible cycles formed by λ' and $\tau(\lambda)$ are negative; hence v_0 will be a negative value (or $-\infty$) in this case.

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

The proof 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.

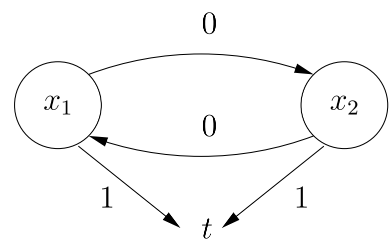

Figure 2: In this controlled LP there is a strategy that is stable, but not optimal. Round vertices are controlled.

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 the controller creates a zero loop, which allows for unboundedness.

7.4 NP∩coNP-Membership

In all the above cases with the property that stability implies optimality, the corresponding decision problem for the Positive Controlled LP-Problem is in NP∩coNP. For a YES-instance, just guess a pure strategy λ and verify that the value of the optimal solution in the resulting LP $S(\lambda)$ is above the threshold.

For a NO-instance, guess a strategy, verify in polynomial time that it is stable and thus optimal and confirm that the corresponding LP has a value below the threshold.

Hence, the univariate discounted, multivariate discounted, and univariate non-discounted cases are all in NP∩coNP.

7.5 The General Positive Controlled LP Case

As we have shown, in many important special cases of Positive Controlled LP-Problem stability implies optimality. This has very nice algorithmic and complexity-theoretic consequences. For the general case however, stability does not imply optimality. We show this by giving a counterexample. Compared to the example in Section 7.3, the one presented here demonstrates that the unboundedness is not essential and we cannot even attain finite optimal values by stable strategies.

Consider the following controlled linear program:

Maximize $v_1 + v_2 + v_3$ subject to

Compulsory constraints:
$$\begin{cases} v_3 \leq v_2 \\ v_3 \leq 2 \end{cases}$$
Optional constraints:
$$\begin{cases} v_1 \leq v_3 \\ v_1 \leq 1 \end{cases}$$
$$\begin{cases} v_2 \leq v_1 \\ v_2 \leq 1 \end{cases}$$

The corresponding graph is shown in Figure 3.

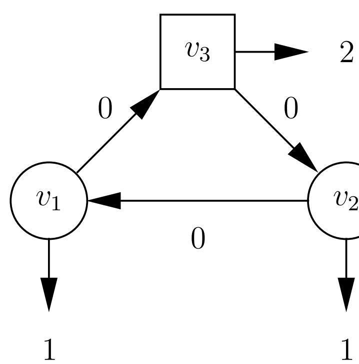

Figure 3: In this controlled LP there is a strategy that is stable but not optimal. Round vertices are controlled.

If the controller selects the constraints $v_1 \leq 1$ and $v_2 \leq 1$, the strategy is stable and the value is 3 (value 1 for each variable). However, if he had selected $v_1 \leq v_3$ and $v_2 \leq v_1$, we would have had the linear program:

Maximize $v_1 + v_2 + v_3$ subject to

$$\begin{cases} v_3 \leq v_2 \\ v_3 \leq 2 \end{cases}$$
$$v_1 \leq v_3$$
$$v_2 \leq v_1,$$

which evaluates to 6 (value two for each variable).

Notice that this example does not contain any hyperedges, i.e., constraints with more than one variable on the right-hand side.

8 NP∩coNP Case: Positive Integer Coefficients

In this section we investigate the controlled LP problem, restricted to have only *nonnegative integers* as coefficients in the constraints, and prove the following result.

Theorem 8.1 *The controlled LP problem with nonnegative integer coefficients belongs to the complexity class* NP∩CO-NP.

For this case, stability does not imply optimality (Section 7.5). Therefore, more advanced reasoning is needed, compared with Section 7.4. Throughout this section we assume that the systems have retreats for both players. This guarantees that there are feasible solutions and that the optimal values are finite. We begin by formally defining cyclic dependencies.

Definition 8.2 *A system S with only one constraint per uncontrolled variable has a* cyclic constraint *over variable v if there is a controller strategy σ such that in $S(\sigma)$ there is a cyclic sequence of controlled variables $v = v_1, v_2, \ldots, v_k = v$ such that for each $i \in \{1, 2, \ldots, k-1\}$, there is a constraint of the form $v_i \leq poly(\overline{y}, v_{i+1}) + w_i$.*[6] *If there is a strategy with at least* two *distinct such sequences, then S has* multiple cyclic constraints *over v.* □

In the sequel we will need the following technical result.

Lemma 8.3 *Suppose that system S has only one constraint for each variable. If S has a cyclic constraint over some variable, then S is either infeasible or unbounded.*

Proof. If S is feasible and has a cyclic constraint over x, then, by combining constraints, we get $x \leq c \cdot x + poly(\mathbf{x} \setminus \{x\}, \mathbf{y})$. For any finite solution to S, the polynomial $poly(\mathbf{x} \setminus \{x\}, \mathbf{y})$ evaluates to some finite number. Since c is a positive integer constant, the value of x can be increased arbitrarily without violating the constraints. □

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

Lemma 8.4 *Given a problem instance S, let σ be an optimal strategy for* MAX. *Then there is a selection τ of exactly one constraint for each non-controlled variable from the tight constraints in $S(\sigma)$, such that in $S(\sigma, \tau)$, the system with only the constraints from σ and τ, there are no cyclic constraints.*

Proof. Let $(\mathbf{x}^*, \mathbf{y}^*)$ be an optimal solution to $S(\sigma)$. Since there are retreats and σ is optimal, $S(\sigma)$ has a feasible solution and $(\mathbf{x}^*, \mathbf{y}^*)$ 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 8.3 the system is either infeasible or unbounded. Either way, we have a contradiction. □

[6] Since the opponent has no choices, each of his constraints $y \leq Q$ can be substituted in the right-hand sides of the controller's constraints and all non-controlled variables can be eliminated. For notational simplicity we assume here that this transformation is already done.

We call a selection τ as in the above lemma an *optimal counterstrategy* for Min.

Lemma 8.5 *Let S be a system with only one constraint per uncontrolled variable, without multiple cyclic constraints and without cyclic constraints with coefficients larger than 1. Let σ be stable in S. Suppose there is a cyclic constraint with sequence $x_0, x_1, \ldots, x_k = x_0$ that* Max *can enforce. Further, assume that by enforcing it, switching from σ* only *in variables in $\{x_1, \ldots, x_k\}$,* Max *can get unbounded values. Call the strategy after switching σ'. If we do potential transformation on S, using the value of each variable under σ as the potentials, then every constraint for a variable $x_i, i \in \{1, \ldots, k\}$ in σ' is homogeneous.*

Proof. For each x_i, let c_i be the constraint for x_i in $S(\sigma')$. Also, let w_i be the constant term in c_i and q_i the value of the rest of the right-hand side of c_i, except x_{i+1}, under σ. First note that no variable except x_{i+1} in any constraint c_i in the cycle can depend on any variable in $\{x_1, \ldots, x_k\}$. If this were the case, there would be multiple cyclic constraints. Suppose that $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ are the variables for which the constraints differ between σ and σ'. For each variable v, let $p(v)$ be its value under σ. For every $j \in \{1, \ldots, r\}$ we have $p(x_{i_j}) \geq w_{i_j} + q_{i_j} + w_{i_j+1} + q_{i_j+1} + \cdots + p(x_{i_{j+1}})$, since σ is stable. By combining such constraints we get $\sum_{i \in \{1,\ldots,k\}} w_i + q_i \leq 0$. Since σ' gives unboundedness, the sum must actually equal 0.

By combining constraints in a similar way we see that for every $j \in \{1, \ldots, r\}$, the value $p(x_{i_j})$ equals $w_{i_j} + q_{i_j} + w_{i_j+1} + q_{i_j+1} + \cdots + p(x_{i_{j+1}})$. The new constant term in the constraint c_{i_j} is $w_{i_j} - p(x_{i_j}) + q_{i_j} + w_{i_j+1} + q_{i_j+1} + \cdots + p(x_{i_{j+1}}) = p(x_{i_j}) - p(x_{i_j}) = 0$. For every other constraint in $\{c_i | 1 \leq i \leq k\}$ the claim follows because they were tight under σ. $\square$

Lemma 8.6 *Assume the same situation as in Lemma 8.5, but switching to σ', enforcing the cyclic constraint,* Max *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 that in the proof of Lemma 8.5. $\square$

Together, Lemmas 8.5 and 8.6 show that any cyclic constraint such that Max can achieve unbounded values by only switching to the constraints associated with the cyclic constraint gets homogeneous constraints after potential transformation with respect to a stable strategy. If there is such a cyclic constraint, it is easy to detect.

Lemma 8.7 *Let S be a system with only one constraint per uncontrolled variable. Let σ and σ' be two strategies such that neither $S(\sigma)$, nor $S(\sigma')$ has any cyclic constraints. Assume that σ is stable. Then σ' is no better than σ.*

Proof. Since $S(\sigma')$ 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 $y \leq w$, for a constant w, in $S(\sigma')$. For each variable v, let $p(v)$ and $p'(v)$ 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 the number of later elements in T that $p'(v) \leq p(v)$ 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 $S(\sigma')$. Since σ is stable we have $w = p'(v) \leq p(v)$.

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

The following procedure can be used to verify that a system S does *not* have a solution with value at least k.

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 thus that σ is optimal.

If these steps can all be computed in polynomial time (as we actually show below), then the problem is in co-NP.

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.

The first thing we can check is whether there is a strategy σ' such that $S(\sigma', \tau)$ has a cyclic constraint over variables $x_1, \dots, x_k$ such that in some constraint $x_i \leq \mathit{poly}(\overline{y}, x_{i+1})$, 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 $S(\tau)$.

The next step is to 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 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 8.6, 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 8.7. Otherwise, we can pick a cyclic constraint, say over $x_1, x_2, \dots, x_k$, and remove all constraints for the variables x_i, and variables who depend on them. This does not affect the other variables. 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. The same must, for the same reasons, hold for any other cyclic constraint. Now by Lemma 8.5, σ is optimal in S. Thus we have short witnesses for NO-instances, and Theorem 8.1 follows.

Corollary 8.8 *The controlled LP problem with all coefficients equal to 0 or larger than 1 belongs to the complexity class* NP∩CO-NP.

9 The General Controlled LP Problem is NP-Hard

We now prove that allowing one negative coefficient in the right-hand side polynomials makes the CONTROLLED LP-PROBLEM NP-hard.

Definition 9.1 *The* GENERAL CONTROLLED LP-PROBLEM *is defined as in Definition 2.1, with the difference that we allow negative coefficients in the polynomials* $p_i^j(\mathbf{y})$ *and* $q_k^l(\mathbf{x})$.

Theorem 9.2 *The* GENERAL CONTROLLED LP-PROBLEM *is NP-hard.*

Proof. We prove NP-hardness by reducing 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 controlled linear program 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 following compulsory constraints:

$$\begin{aligned} x_i &\leq 1, \\ \bar{x}_i &\leq 1 - x_i. \end{aligned}$$

For every clause $C'_j = l_1 \vee l_2 \vee l_3$ in ϕ add the following optional constraint:

$$\left\{ \begin{aligned} C_j &\leq l_1, \\ C_j &\leq l_2, \\ C_j &\leq l_3. \end{aligned} \right.$$

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 the LP 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.

Assume ϕ is satisfiable. There must exist a variable assignment that assures that every clause is satisfied, i.e., $\forall j : C_j = 1$, which implies $\sum_{j=1}^m C_j = m$. Thus the optimal solution to the LP program is $n + m$. □

10 Controlled LP Applications

10.1 Simple Stochastic Games

In this section we show how controlled LPs express *simple stochastic games* (SSG), a subclass of stochastic games introduced and studied by Shapley [21] (see [7, 8, 10] 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 and 1 are called the 0-*sink* and the 1-*sink*, and the vertex set V is partitioned into subsets V_1, V_0, V_{avg} of the maximizing player 1, minimizing player 0, 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 1 wants to maximize and player 0 to minimize the probability of getting to the 1-sink. SSGs turn out to possess a *value* ν and are solvable in positional strategies, optimal for both players, securing them the equilibrium probability ν of getting to the 1-sink.

The natural decision problem consists in determining, given an SSG and a probability $p \in [0, 1]$, whether the value of the game is $\geq p$. Condon [7] showed that this problem is in NP∩coNP, and suggested several algorithms for finding values and optimal strategies.

Actually, the first step Condon [7] applies is polynomial reduction[7] 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 [7, 22] for details), every edge (u, v) with assigned probability p in the original game[8] 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 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 optional constraints $x \leq p \cdot y$ for every edge (x, y) from V_1, and compulsory constraints $y \leq p \cdot x$ for every edge from V_0, 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 $u \leq \sum_{i \in I} p_i \cdot v_i$ (clearly, $\sum_{i \in I} p_i < 1$ 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 7.2[9]. 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.[10]

[7] Corrected in [22] to become polynomial.

[8] Edges of players 0 and 1 are given probability 1.

[9] Actually, Shapley in the original definition of SGs stipulated discountedness; see [12].

[10] One may argue that such a quadratic blow-up in the number of vertices may only happen when there are quadratically many edges, and therefore, Ludwig's algorithm [14] remains *subexponential in the length of input*. More precisely, let n and $m = \Theta(n^2)$ be the numbers of vertices and edges in the original game; hence

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. As an additional benefit, we also have interior point algorithms available.

10.2 Discounted Payoff Games

In this section we show how to describe Discounted Payoff Games as univariate discounted Positive Controlled LP-Problem. Let the game $G = (V = V_{\text{MIN}} \cup V_{\text{MAX}}, E, w, \delta)$ be given; see [22] for definitions. The objective of the adversaries is to maximize/minimize

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

and it has been shown that the optimal value vector of a DPG is the unique solution to the set of equations [22]:

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

It is straightforward to see that this corresponds to the following controlled LP:

Maximize $\sum_i v_i$ subject to

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

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

This controlled LP is exactly of the form discussed in Section 7.1, hence stability implies optimality. Any stable vector in the linear program will also satisfy the system of equations determining the optimal vector of a Discounted Payoff Game. Therefore, a DPG can be expressed as a univariate discounted Positive Controlled LP-Problem.

---

the size of the input graph is $|G| = \Theta(n^2)$. Reducing to the game of outdegree 2 results in $n' = \Theta(n^2)$ vertices, and Ludwig's algorithm solves such a game in $2^{O(\sqrt{n'})} = 2^{O(n)} = 2^{o(|G|)}$ time, subexponential in input size $|G|$. This argument is, however, faulty, because in the scenario described above *any* straightforward algorithm is subexponential in the length of input. Indeed, there may be at most $n^n = 2^{n \log n} = 2^{o|G|}$ strategies, hence *every* brute-force search for the best strategy in this space does the job in time subexponential in $|G|$. Therefore, a more adequate meaning of *subexponential* in this context should be given/estimated in the *number of vertices*, rather than the input size. In particular, in the above scenario Ludwig's algorithm should be considered *exponential* in n. In Section 11 we give precise bounds for our algorithms in terms of the numbers of vertices n and edges m, like $2^{O(\log m \cdot \sqrt{n/\log n})}$ and $2^{O(\sqrt{n \log(m/\sqrt{n})} + \log m)}$, *subexponential* in n whenever $m = \Theta(poly(n))$.

10.3 The Longest Shortest Paths

The Longest Shortest Paths Problem (LSP) [5] consists in finding, for a given weighted digraph with a sink and a set of controlled vertices, a selection of exactly one edge from each controlled vertex maximizing the lengths of the shortest paths from all vertices to the sink.

An instance I of the LSP problem reduced to two instances L^- and L^+ of the univariate non-discounted Positive Controlled LP-Problem. The need for two instances arises from the different interpretations of zero-weight loops in the LSP and controlled LP. Such loops give zero distance in the LSP, but yield unbounded values of the associated variables in the LP.

The two instances L^- and L^+ are obtained by first applying to I the transformations $\Delta^{-1}_{\times n}$ and $\Delta^{+1}_{\times n}$, respectively, from Section 7.3.

Afterward, to get L^- and L^+ write optional constraints

$$x \leq w_{(x,y)} + y$$

for each controlled vertex with successor y via edge (x, y) of weight $w_{(x,y)}$. Write compulsory constraints

$$y \leq w_{(y,x)} + x$$

for each uncontrolled vertex y. Add the constraint $t \leq 0$ for the sink and the "retreat" constraints $x \leq -M$ for each controlled vertex, where $M > 0$ is big enough.

In L^- every nonpositive loop from I becomes negative, and every positive loop remains positive. Starting with the admissible strategy "retreat everywhere", run any version of iterative improvement by attractive switches on L^-, throwing away unbounded variables and constraints, as described in Section 3.1. To find out the original value for finite paths, we divide by n and round up to the nearest integer.

After solving L^- we know that all vertices corresponding to unbounded variables are part of positive loops in the LSP instance I. All vertices corresponding to finite values bigger than zero have also chosen the optimal strategy and obtained optimal values. If the finite values are less then zero then it is possible that there exists a zero-weight loop in I, which was not found in L^-.

To find out whether this is the case, we solve the second instance L^+, and for every variable with a finite value less than zero in L^- we check whether its value in L^+ is unbounded or not. If it is, we know that such variable have value 0 in I.

All previous results and algorithms now apply to the LSP problem.

10.4 Mean Payoff Games

Mean Payoff Games are adversary two-player full information infinite games played on finite, directed, edge-weighted, leafless graphs. The objective of the players is to maximize/minimize, in the limit, the sum of the edge weights divided by the number of edges traversed. These games are interesting in their own right but are also motivated by the fact that Parity Games (and thus model checking) reduce to them. For details see [17, 9, 11, 20, 6]. Many important decision and optimization problems are reducible [4] to the Longest Shortest Paths problem, discussed in Section 10.3. Therefore it follows directly that they can be expressed by the univariate non-discounted Positive Controlled LP-Problem.

11 Algorithms

11.1 Single and Multiple Switch Algorithms

In Section 3 we discussed a generic switching algorithm for the Positive Controlled LP-Problem. There are several switching schemes that can be employed in this template algorithm, and we describe some of them briefly here. For more detailed descriptions and applications of these schemes see [1].

- *Single Random Switch*: select uniformly at random exactly one out of the attractive switches in each iteration.
- *Single Greedy Switch*: select "greedily" exactly one 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*: flip a coin for every attractive switch in each iteration (or uniformly at random choose a subset of all available attractive switches.)

11.2 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 [13]. Matoušek, Sharir, and Welzl independently proved the same result for another algorithm [15]. In 1995, Ludwig was the first to propose using these techniques in a game-theoretic setting [14], but in a restricted form, only for games on graphs of outdegree 2.

11.2.1 Kalai-Style Randomization for Controlled LP

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

Given an instance S of PCLP, 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 PCLP proceeds as follows (cf., [2, 3]). Given an instance S and a strategy σ, it applies recursively to find a big enough set $\mathcal{S}$ of *better* subproblems (facets) that 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 Positive Controlled LP-Problem (whenever stability implies optimality; otherwise we find a local optimum in subexponential time), where n and m are the numbers of controlled variables and optional constraints, respectively. If we assume that m is polynomially bounded in n, the bound is $2^{O(\sqrt{n \log n})}$.

11.2.2 Matoušek-Sharir-Welzl-Style Randomization for Controlled LP

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

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

The Switch function takes a strategy σ and an optional constraint $x \le e$ and returns the strategy obtained by changing the choice of σ for x to $x \le e$. In combinatorial optimization terms, line 3 of the algorithm corresponds to selecting a facet in the space of positional strategies. Line 4 corresponds to recursive optimization on the rest of this space.

The expected number of iterations of this algorithm is $2^{O(\sqrt{n \log(m/\sqrt{n})} + \log m)}$, where n and m are the numbers of controlled variables and optional constraints, respectively, similar to the bound for Kalai's algorithm. Again, if m is polynomial in n, the bound is $2^{O(\sqrt{n \log n})}$.

12 Controlled Max Bipartite Matching and Max Flow

Previous sections show that a number of problems that belong to P have controlled versions in NP∩co-NP, but also that the general controlled LP problem (where negative coefficients are allowed) is NP-complete. It actually seems that the latter case is more common, and here we give two additional examples.

In the Controlled Max Bipartite Matching problem, we are given a bipartite, undirected graph $G = (V = (A \cup B), E)$, a subset $C \subseteq V$ 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.

The Controlled Max Bipartite Matching 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* positive constraint we use in PLCP. Skew symmetry and flow preservation constraints require negative coefficients. Actually, no matter which representation tricks one tries to use, negative coefficients are unavoidable. Indeed, Controlled Maximum Flow is also NP-hard.

In the Controlled Maximum Flow 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 reduction from CNF-Non-Validity; the problem of determining whether a given CNF formula is *not* a tautology. CNF-Non-Validity is NP-complete, since CNF-Validity is co-NP-complete.

13 Conclusions

We briefly enumerate several problems remaining to be investigated.

1. Interior point schemes should be investigated further, as per their potential advantage over discrete schemes. 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 CONTROLLED LP-PROBLEM.

2. For many special cases, local search algorithms, similar to those for the related games, give optimal results. In the general case, local search does not give optimal solutions. In the case of integer coefficients, however, we could still prove NP∩coNP-membership. Can this problem also be solved in expected subexponential time?

3. Does the general positive controlled LP problem belong to coNP?

4. The positive controlled LP problem catches many important optimization problems, including several games. Are there any other interesting optimization problems of this kind?

5. Can the subexponential upper bound for the problem be improved to $2^{o(\sqrt{n \log n})}$, or even to a polynomial one?

References

[1] 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.

[2] 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.

[3] 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/`.

[4] H. Björklund, S. Sandberg, and S. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. Technical Report DIMACS-2004-05, DIMACS: Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University, NJ, March 2004. `http://dimacs.rutgers.edu/TechnicalReports/`.

[5] 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. Preliminary full version: TR DIMACS-2004-05 `http://dimacs.rutgers.edu/TechnicalReports/`.

[6] H. Björklund, S. Sandberg, and S. Vorobyov. Memoryless determinacy of parity and mean payoff games: A simple proof. *Theoretical Computer Science*, 310(1-3):365–378, 2004.

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

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

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

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

[11] 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.

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

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

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

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

[16] J. Matoušek, M. Sharir, and M. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.

[17] H. Moulin. Extensions of two person zero sum games. *J. Math. Analysis and Applic.*, 55:490–508, 1976.

[18] 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.

[19] 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/`.

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

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

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