Three views of LP-type optimization problems

Jiří Matoušek

Department of Applied Mathematics and
Institute of Theoretical Computer Science (ITI)
Charles University
Malostranské nám. 25, 118 00 Praha 1
Czech Republic

Petr Škovroň

Department of Applied Mathematics
Charles University
Malostranské nám. 25, 118 00 Praha 1
Czech Republic

September 10, 2003

Abstract

An axiomatically defined class of optimization problems, called *LP-type problems* (or also *generalized linear programming problems*) and introduced by Sharir and Welzl, includes linear programming, the smallest enclosing ball for a given point set in $\mathbf{R}^d$, and many other important problems. We investigate mathematical properties of LP-type problems. In particular, we introduce two axiomatic definitions, the *concrete LP-type problems* and the *acyclic violator spaces*, and we show that they are exactly equivalent to the original definition, thus providing two new approaches to LP-type problems.

1 Introduction

Sharir and Welzl [10] developed a randomized variant of the simplex algorithm for linear programming. It constituted an important progress in the theory of linear programming, as it was later shown to be subexponential [9]. Together with a similar result independently obtained by Kalai [7], this was the first linear programming algorithm provably requiring number of arithmetic operations subexponential in the dimension and number of constraints (independent of the precision of the input numbers).

Abstract LP-type problems. Another great advantage of the Sharir–Welzl algorithm was a fairly abstract formulation. They introduced an axiomatically defined class of optimization problems, called *LP-type problems*, which includes linear programming (in a suitable formulation), but also a number of other significant problems, such as the minimum ball enclosing a given point set in $\mathbf{R}^d$, the minimum-volume ellipsoid enclosing a given point set in $\mathbf{R}^d$, or the distance of two convex polytopes in $\mathbf{R}^d$. Also see [9], [2], [1] for many other concrete examples of LP-type problems, [4] for an algorithm in game theory related to LP-type problems, and [3] for a beautiful mathematical application. Once it is shown that a particular optimization problem is an LP-type problem, and certain algorithmic primitives are implemented for it, several efficient algorithms are at disposal immediately: the Sharir–Welzl algorithm, another randomized optimization algorithm due to Clarkson [6] (see [5] for discussion of how it fits the LP-type framework), a deterministic version of it [5], an algorithm for computing the minimum solution that violates at most k of the given n constraints [8], and probably more are to come in the future.

An LP-type problem is given by a finite set H of *constraints* and a *value* $w(G)$ for every subset $G \subseteq H$. The values can be real numbers or, for technical convenience, elements of any other linearly ordered set. Intuitively, $w(G)$ is the minimum value of a solution that satisfies all constraints in G. The assignment $G \mapsto w(G)$ has to obey the axioms in the following definition.

Definition 1.1 *An* (abstract) LP-type problem *is a fourtuple* $(H, w, W, \leq)$*, where* H *is a (finite) set,* W *is a set linearly ordered by* $\leq$*, and* $w\colon 2^H \to W$ *is a mapping satisfying the following two conditions:*

(M) *for all* $F \subseteq G \subseteq H$ *we have* $w(F) \leq w(G)$ *("monotonicity"), and*

(L) *for all* $F \subseteq G \subseteq H$ *and all* $h \in H$ *with* $w(F) = w(G)$ *and* $w(G) < w(G \cup \{h\})$*, we have* $w(F) < w(F \cup \{h\})$ *("locality").*

As our running example, we will use the smallest enclosing ball problem, where H is a finite point set in $\mathbf{R}^d$ and $w(G)$ is the radius of the smallest ball that encloses all points of G. In this case monotonicity is obvious, while verifying locality requires a nontrivial geometric result (such as the uniqueness of the smallest enclosing ball for every set).

Concrete LP-type problems. Although intuitively one thinks about $w(G)$ as the value of an optimal solution of an optimization problem, the solution itself is not explicitly represented in Definition 1.1. In specific geometric examples, the constraints can usually be interpreted as subset of some ground set X of points, and the optimal solution for G is the point with the smallest value in the intersection of all constraints in G. For example, in linear programming, the constraints are halfspaces, the value is given by the objective function, and the optimum is the point with minimum value in the admissible region, i.e., the intersection of the halfspaces. In order to have a unique optimum for every set of constraints (which is needed for (L), the locality axiom), one assumes that the points are linearly ordered by the value; for linear programming, we can always take the lexicographically smallest optimal solution, for instance.

Such an intepretation is possible for the smallest enclosing ball problem too, although it looks a bit artificial. Namely, the "points" of X are all balls in $\mathbf{R}^d$, and the ordering can be an arbitrary linear extension of the partial ordering of balls by radius.

The following definition captures this approach to LP-type problems.

Definition 1.2 *A* concrete LP-type problem *is a triple* $(X, \preceq, \mathcal{H})$*, where* X *is a set linearly ordered by* $\preceq$*,* $\mathcal{H} \subseteq 2^X$ *is a finite multiset, and for any* $\mathcal{G} \subseteq \mathcal{H}$*, the intersection* $\bigcap \mathcal{G}$ *is nonempty and has a minimum element with respect to* $\preceq$ *(for* $\mathcal{G} = \emptyset$ *we define* $\bigcap \mathcal{G} = X$*).*

Given any concrete LP-type problem $\mathcal{P} = (X, \preceq, \mathcal{H})$, we obtain an abstract LP-type problem $P = (\mathcal{H}, w, X, \preceq)$ according to Definition 1.1 by putting $w(\mathcal{G}) = \min\{\bigcap \mathcal{G}\}$, as is easy to check (proof omitted).

A little surprising is the converse, which we will prove below: Any abstract LP-type problem $P_0 = (H, w, W, \leq)$ has a "concrete representation," that is, a concrete LP-type problem $\mathcal{P}$ such that the LP-type problem P obtained from $\mathcal{P}$ as above is isomorphic to P_0 in a natural sense.

LP-type problems with $-\infty$. The original definition of LP-type problems in [10] is slightly more general than Definition 1.1. It allows for breaking locality under certain conditions: W has a special minimal element $-\infty$, and locality need not hold if $w(F) = w(G) = -\infty$. This special role of $-\infty$ is useful, for example, to tame the usual linear programming problem to become an LP-type problem. One can also think of the LP-type problems with $-\infty$ as "partial" LP-type problems, where sets of constraints with values below certain threshold are ignored. All of the current algorithms for

LP-type problems work for LP-type problems with $-\infty$ as well under a mild additional assumption (since, roughy speaking, once they start with a value larger than $-\infty$, they always proceed upwards to larger values). On the other hand, many natural problems do not need $-\infty$ at all, and for linear programming it can be avoided by restricting the admissible solutions to the positive orthant, say.

A concrete LP-type problem always yields an abstract LP-type problem satisfying locality everywhere, and so LP-type problems with $-\infty$ cannot have a concrete representation in the above sense. It would be interesting to find some other kind of "concrete representation," and in general to investigate how much the LP-type problems with $-\infty$ may really differ from those without $-\infty$.

Violator spaces. Let $(H, w, W, \leq)$ be an abstract LP-type problem. It is natural to define that a constraint $h \in H$ *violates* a set $G \subseteq H$ of constraints if $w(G \cup \{h\}) > w(G)$. For example, in the smallest enclosing ball problem, a point h violates a set G if it lies outside of the smallest ball enclosing G (which is unique).

Let us write $\mathsf{V}(G) = \{h \in H : w(G \cup \{h\}) > w(G)\}$ for the set of all constraints violating G. It turns out that the knowledge of $\mathsf{V}(G)$ for all $G \subseteq H$ is enough to describe the "structure" of an LP-type problem. That is, while we cannot reconstruct W, $\leq$, and w from this knowledge, it is natural to consider two LP-type problem with the same mapping $\mathsf{V}: 2^H \to 2^H$ the same (isomorphic). Indeed, the algorithmic primitives needed for implementing the Sharir–Welzl algorithm and the other algorithms for LP-type problems mentioned above can be phrased in terms of testing violation (does $h \in \mathsf{V}(G)$ for a certain set $G \subseteq H$?), and they never deal explicitly with the values of w.

We now introduce the notion of *violator space*:

Definition 1.3 *A* violator space *is a pair* (H, V)*, where* H *is a finite set and* V *is a mapping* $2^H \to 2^H$ *such that*

(VC) $G \cap \mathsf{V}(G) = \emptyset$ *holds for all* $G \subseteq H$ *("consistency"), and*

(VL) *for all* $F \subseteq G \subseteq H$*, where* $G \cap \mathsf{V}(F) = \emptyset$*, we have* $\mathsf{V}(G) = \mathsf{V}(F)$ *("locality").*

Two violator spaces (H, V) and (H', V') are called *isomorphic* if they differ only by renaming of the elements of H; that is, if there is a bijection $\varphi: H \to H'$ such that $\mathsf{V}'(\varphi(G)) = \varphi(\mathsf{V}(G))$ for all $G \subseteq H$.

As we will check in Section 2, given any abstract LP-type problem $(H, w, W, \leq)$, the pair (H, V) with $\mathsf{V}(G) = \{h \in H : w(G \cup \{h\}) > w(G)\}$ is a violator space. However, there are violator spaces that cannot be obtained from any LP-type problem.

In Section 3 we define the notion of acyclicity of violator spaces. We will see that a violator space constructed from an LP-type problem is always acyclic. Moreover, it turns out that this property characterizes the violator spaces obtained from LP-type problems, and thus any acyclic violator space can be represented as an LP-type problem (abstract or concrete). This, together with the equivalence of abstract and concrete LP-type problem, is stated in the next theorem.

Theorem 1.4 *Let (H, V) be an acyclic violator space. Then there exists a concrete LP-type problem such that the violator space arising from it (that is, first we construct the abstract LP-type problem as described below Definition 1.2, and then we define V as the violators mapping of that abstract LP-type problem) is isomorphic to (H, V).*

The construction is illustrated on simple instances of problems of linear programming and the smallest enclosing ball.

Additional results. Several more results concerning violator spaces have been achieved in the MSc. thesis of the second author [11]. These are summarized, without proofs, in Section 5.

2 Preliminaries on abstract LP-type problems

First we recall the notion of a basis in an abstract LP-type problem [10]:

Definition 2.1 ([10]) *Given an LP-type problem $(H, w, W, \leq)$, we say that $B \subseteq H$ is a* basis *if for all proper subsets $F \subset B$ we have $w(F) < w(B)$.*

Now we prove several useful facts.

Lemma 2.2 *Consider an LP-type problem $(H, w, W, \leq)$. Let $A, B \subseteq H$, where B is not violated by any $h \in A$ ($A \cap \mathsf{V}(B) = \emptyset$). Then $w(A \cup B) = w(B)$.*

Proof. From monotonicity, we immediately obtain the inequality "$\geq$".

The inequality "$\le$" can be shown by induction on $|A|$. If $|A| = 1$, i.e., $A = \{h\}$, then $w(B \cup \{h\}) > w(B)$ would imply that B is violated by $h \in A$, which is impossible.

Let $|A| > 1$ and $A = A_0 \mathbin{\dot\cup} \{h\}$ (disjoint union). From the induction hypothesis we have $w(A_0 \cup B) = w(B)$. Now, if $w(B \cup A_0) < w(B \cup A_0 \cup \{h\})$, then by locality (for $B \cup A_0$, B and h) we get $w(B) < w(B \cup \{h\})$, i.e., $h \in A \cup \mathsf{V}(B) = \emptyset$, which is not possible. So $w(B) = w(B \cup A_0) \ge w(B \cup A_0 \cup \{h\}) = w(B \cup A)$. We have proved $w(B) \ge w(A \cup B)$. □

We want to express all important properties of an LP-type problem using only V and to live happily without w. The following statement is the first step in this direction.

Lemma 2.3 *Given an LP-type problem $(H, w, W, \le)$ and a set $M \subseteq H$, the following conditions are equivalent:*

(a) *M is a basis (for all $N \subset M$ we have $w(N) < w(M)$).*

(b) *For any proper subset $N \subset M$ we have $(M \setminus N) \cap \mathsf{V}(N) \ne \emptyset$.*

Proof. Having $N \subset M$, we want to find $h \in (M \setminus N) \cap \mathsf{V}(N)$, i.e., h such that $w(N) < w(N \cup \{h\})$.

We suppose for contradiction that for every $h \in M \setminus N$ we have $h \notin \mathsf{V}(N)$. Then by Lemma 2.2 we have $w(M) = w((M \setminus N) \cup N) = w(N)$, which is not the case. This shows that (a) implies (b).

Now we suppose that (b) holds and that $N \subset M$. Then $M \setminus N$ and $\mathsf{V}(N)$ share some element h. We get $w(N) \le w(M \setminus \{h\}) < w(M)$. □

Lemma 2.4 *Consider an LP-type problem $(H, w, W, \le)$. Then for any $A, B \subseteq H$ with $\mathsf{V}(A) = \mathsf{V}(B)$ we have $w(A) = w(B)$. Conversely, $w(A) = w(B) = w(A \cup B)$ implies $\mathsf{V}(A) = \mathsf{V}(B)$. In particular, if $A \subseteq B$ and $w(A) = w(B)$, then $\mathsf{V}(A) = \mathsf{V}(B)$.*

Note that the condition $w(A) = w(B)$ generally does not suffice for $\mathsf{V}(A) = \mathsf{V}(B)$. For example, having any H, we can define w by $w(G) = |G|$ for all $G \subseteq H$; then any G's of the same size have the same w, however, $\mathsf{V}(G) = H \setminus G$, and so no distinct G's share the value of V. Rougly speaking, the equality $w(G_1) = w(G_2)$ may hold just "by accident". Among others, this shows that w really does not reflect the combinatorial structure of the problem in a natural way.

Proof of Lemma 2.4. Let $w(A) \neq w(B)$. Without loss of generality we assume $w(A) < w(B)$ (note that here we use the linearity of ordering $\leq$). If $B \cap \mathsf{V}(A) = \emptyset$, from Lemma 2.2 we would get $w(A \cup B) = w(A)$, which contradicts $w(A \cup B) \geq w(B) > w(A)$. So there necessarily exists $x \in B \cap \mathsf{V}(A)$, but since $x \in B$, we have $x \notin \mathsf{V}(B)$. So $\mathsf{V}(A) \neq \mathsf{V}(B)$.

Conversely, suppose $w(A) = w(B) = w(A \cup B)$. We want to show $\mathsf{V}(A) = \mathsf{V}(B)$, i.e., that $w(A) < w(A \cup \{h\})$ holds iff $w(B) < w(B \cup \{h\})$ holds. By symmetry, it suffices to show only one of the implications. We assume $w(A) < w(A \cup \{h\})$. Then $w(A \cup B) = w(A) < w(A \cup \{h\}) \leq w(A \cup B \cup \{h\})$. Since $B \subseteq A \cup B$ and $w(B) = w(A \cup B)$, we may use locality, which gives $w(B) < w(B \cup \{h\})$. So the desired equivalence holds. $\square$

The following statement will help us express the condition of locality without w.

Lemma 2.5 *Consider an LP-type problem $(H, w, W, \leq)$. For sets $F \subseteq G \subseteq H$, $w(F) = w(G)$ is equivalent to $G \cap \mathsf{V}(F) = \emptyset$.*

Proof. If $w(F) = w(G)$, then $\mathsf{V}(F) = \{h : w(G) < w(F \cup \{h\})\}$ and $G \cap \mathsf{V}(F) = \{h \in G : w(G) < w(F \cup \{h\})\}$. Since $F \subseteq G$ is given and we look for h only among elements of G, we have $F \cup \{h\} \subseteq G$, by monotonicity $w(G) \geq w(F \cup \{h\})$, and $G \cap \mathsf{V}(F) = \emptyset$.

Conversely, suppose $G \cap \mathsf{V}(F) = \emptyset$. Then for any $h \in G$ we have $h \notin \mathsf{V}(F)$, so $w(F \cup \{h\}) = w(F)$. Then, by Lemma 2.2, we get $w(F \cup G) = w(F)$. Since $F \subseteq G$, we have $F \cup G = G$, and so $w(G) = w(F)$ indeed. $\square$

The following lemma states properties of the mapping V.

Lemma 2.6 *Consider an LP-type problem $(H, w, W, \leq)$. The violators mapping V has the following properties:*

- *for all $G \subseteq H$, we have $G \cap \mathsf{V}(G) = \emptyset$, and*
- *for all $F \subseteq G \subseteq H$ such that $G \cap \mathsf{V}(F) = \emptyset$, we have $\mathsf{V}(G) = \mathsf{V}(F)$.*

That is, (H, V) is a violator space.

Proof. Clearly $G \cap \mathsf{V}(G) = \emptyset$, since $w(G) = w(G \cup \{g\})$ for any $g \in G$.

If $G \cap \mathsf{V}(F) = \emptyset$, then by Lemma 2.5 we get $w(F) = w(G)$, and from Lemma 2.4 we have $\mathsf{V}(G) = \mathsf{V}(F)$. $\square$

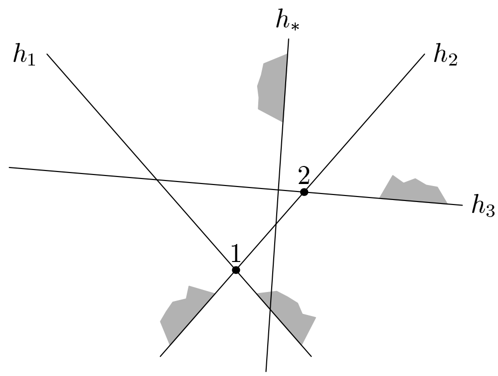

Figure 1: A linear programming example.

At first glance, one might think that for $F \subseteq G$ we should have $\mathsf{V}(F) \supseteq \mathsf{V}(G)$. Unfortunately, this is not the case, as the linear programming example in Fig. 1 shows (the y-coordinate is to be minimized).

We put $F = \{h_1, h_2\}$ and $G = \{h_1, h_2, h_3\} \supseteq F$. The point 1 is minimum in the intersection of F, and 2 is minimum in the intersection of G. We have $1 \in h$, $2 \notin h$, and so $h \notin \mathsf{V}(F)$ and $h \in \mathsf{V}(G)$.

3 Violator spaces

In this section we prove Theorem 1.4 (every acyclic violator space originates in some concrete LP-type problem). Given an acyclic violator space, we need some appropriate linearly ordered set X of points, and then we will identify the elements of H with certain subsets of X.

What set X will we take? A promising suggestion is to take the set of bases; however, it turns out that one has to make some of the bases equivalent.

So we will proceed in the following way. We will define the bases of a violator space and the equivalence of bases, then we let X be the set of the equivalence classes, and we will order X (if possible). Then we will assign subsets of X to the elements of H, and this will complete the construction of the concrete LP-type problem.

The following definition is justified by Lemma 2.3.

Definition 3.1 *Consider a violator space* (H, V). *We say that a set* $B \subseteq H$ *is a* basis *if for all proper subsets* $F \subset B$ *we have* $B \cap \mathsf{V}(F) \neq \emptyset$.

Having $M \subseteq H$, *we say that* $B \subseteq H$ *is a* basis of M *if*

- B *is a basis,*
- $B \subseteq M$, *and*
- $M \cap \mathsf{V}(B) = \emptyset$.

The set of all bases in the violator space (H, V) will be denoted by $\mathcal{B}$.

The expected properties of bases are summarized in the following statement.

Observation 3.2

(i) *In a violator space* (H, V), *every set* M *has a basis.*

(ii) *Every basis is a basis of itself.*

(iii) *If* B *is a basis of* M, *then* $\mathsf{V}(M) = \mathsf{V}(B)$.

Proof. To prove (i), it suffices to take a set $B \subseteq M$ minimal with respect to inclusion for which $\mathsf{V}(B) \cap M = \emptyset$. To show that B is a basis of M, it suffices to show that B is a basis. For contradiction, assume we have a proper subset $A \subset B$ for which $\mathsf{V}(A) \cap B = \emptyset$. From locality we get $\mathsf{V}(A) = \mathsf{V}(B)$. But then $A \subseteq M$ and $\mathsf{V}(A) \cap M = \mathsf{V}(B) \cap M = \emptyset$, which contradicts the minimality of B.

The statement (ii) is immediate.

If B is a basis of M, we have $M \cap \mathsf{V}(B) = \emptyset$, and since $B \subseteq M$, we get $\mathsf{V}(B) = \mathsf{V}(M)$ directly from locality. □

The following lemma is a foundation for the definition of the equivalence of bases.

Lemma 3.3 *For bases* B, C *in a violator space* (H, V), *the following statements are equivalent:*

(a) *There exists an* $M \subseteq H$ *such that both* B *and* C *are bases of* M.

(b) *Both* B *and* C *are bases of* $B \cup C$.

(c) $\mathsf{V}(B) = \mathsf{V}(C)$.

Proof. The implication (b) $\Rightarrow$ (a) is clear—we can put $M = B \cup C$. Assuming (a), Observation 3.2 gives $\mathsf{V}(B) = \mathsf{V}(M) = \mathsf{V}(C)$., i.e., (c) holds.

Now we suppose (c), and we will show that B is a basis of $B \cup C$ (the proof for C is symmetric). To show this, it suffices to prove that $(B \cup C) \cap \mathsf{V}(B) = \emptyset$. From consistency we have $C \cap \mathsf{V}(C) = \emptyset$, and using $\mathsf{V}(B) = \mathsf{V}(C)$ we get $C \cap \mathsf{V}(B) = \emptyset$. Together with $B \cap \mathsf{V}(B) = \emptyset$ (consistency again), this immediately gives $(B \cup C) \cap \mathsf{V}(C) = \emptyset$. So (c) implies (b), which finishes the proof. □

Definition 3.4 *If the conditions in Lemma 3.3 are fulfilled, we say that B and C are* equivalent *($B \sim C$).*

The relation $\sim$ defined on the set of all bases is an equivalence relation (this follows from part (c) in Lemma 3.3). The class of the equivalence $\sim$ containing a basis B will be denoted by $[B]$.

Now we are going to define an ordering of the bases, an ordering of the equivalence classes, and the (long awaited) notion of acyclicity of a violator space.

Definition 3.5 *For $F, G \subseteq H$ in a violator space (H, V), we say that $F \leq_0 G$ (F is* locally smaller *than G) if $F \cap \mathsf{V}(G) = \emptyset$.*

For equivalence classes $[B], [C] \in \mathcal{B}/\!\sim$, we say that $[B] \leq_0 [C]$ if there exist $B' \in [B]$ and $C' \in [C]$ such that $B' \leq_0 C'$.

We define the relation $\leq_1$ on the equivalence classes as the transitive closure of $\leq_0$. The relation $\leq_1$ is clearly reflexive and transitive. If it is antisymmetric, we say that the violator space is acyclic*, and we define the relation $\leq$ as an arbitrary linear extension of $\leq_1$.*

Note that in definition of $[B] \leq_0 [C]$ we do not require $B' \leq_0 C'$ to hold for *every* B' and C'.

Proof of Theorem 1.4. We are given an acyclic violator space (H, V) and $\mathcal{B}$ and $\sim$ are as above. We define the mapping $S\colon H \to 2^{\mathcal{B}/\sim}$ that will act as a "concretization" of the constraints in H:

$$S(h) = \{[B] : B \in \mathcal{B},\, h \notin \mathsf{V}(B)\}.$$

Further, let $\mathcal{H}$ be the image of the mapping S (taken as a multiset), i.e.,

$$\mathcal{H} = \{S(h) : h \in H\}.$$

Thus, S is a bijection between H and $\mathcal{H}$. Let σ be the induced bijection of 2^H and $2^{\mathcal{H}}$ defined by $\sigma(G) = \{S(h): h \in G\}$.

Consider the triple $(\mathcal{B}/\sim, \leq, \mathcal{H})$. This is a concrete LP-type problem; the only thing to check is that every intersection $\bigcap \mathcal{G}$ is nonempty. Let B_0 be a basis of H; then $\mathsf{V}(B_0) = \emptyset$ and $[B_0] \in S(h)$ for any $h \in H$. So any intersection $\bigcap \mathcal{G}$ contains $[B_0]$.

Now we define a mapping $w: 2^H \to \mathcal{B}/\sim$ as follows: $w(M) = [B]$, where B is any basis of M. The key property of w is this:

Lemma 3.6 *For any $G \subseteq H$ and $h \in H$, $w(G) < w(G \cup \{h\})$ is equivalent to $h \in \mathsf{V}(G)$.*

Proof. Using monotonicity we obtain that $w(G) < w(G \cup \{h\})$ is equivalent to $w(G) \neq w(G \cup \{h\})$, which by the definition of w means that no basis of G is equivalent to any basis of $G \cup \{h\}$; i.e., $\mathsf{V}(G) \neq \mathsf{V}(G \cup \{h\})$. By locality in the violator space (H, V), this is equivalent to $\mathsf{V}(G) \cap (G \cup \{h\}) \neq \emptyset$. However, since $\mathsf{V}(G) \cap G = \emptyset$, necessarily $\mathsf{V}(G) \cap \{h\} \neq \emptyset$, and so $h \in \mathsf{V}(G)$. □

Furthermore, w is exactly the cost function of the concrete problem (up to renaming elements of H):

Lemma 3.7 *The value function of the concrete LP-type problem $(\mathcal{B}/\sim, \leq, \mathcal{H})$ is $w \circ \sigma^{-1}$.*

Proof. We want to prove that the value function $\widehat{w}$ defined as $\widehat{w}(\mathcal{G}) = \min \bigcap \mathcal{G}$ is equal to the mapping $w \circ \sigma^{-1}$. We will prove that $\widehat{w} \circ \sigma(G) = w(G)$. For the proof we fix the value of G, define $\mathcal{G} = \sigma(G)$, and let B be any basis of G (so $w(G) = [B]$). For proving $\widehat{w}(\mathcal{G}) = w(G)$, we will show two inequalities.

We start with $\leq$: First we show that $[B] \in \bigcap \mathcal{G}$. For contradiction, suppose $[B] \notin \bigcap \mathcal{G}$. Then $[B] \notin S(h_0)$ for some $h_0 \in G$, i.e., $[B] \notin \{[A]: h_0 \notin \mathsf{V}(A)\}$. So $h_0 \in \mathsf{V}(B) = \mathsf{V}(G)$, which contradicts $h_0 \in G$. So we see that $[B] \in \bigcap \mathcal{G}$, and $\widehat{w}(\mathcal{G}) = \min \bigcap \mathcal{G} \leq [B] = w(G)$.

Now it remains to show the second inequality; i.e., $\widehat{w}(\mathcal{G}) = \min \bigcap \mathcal{G} \geq [B] = w(G)$. For all $[C]$ in $\bigcap \mathcal{G}$ we have to show that $[C] \geq [B]$. Since $\bigcap \mathcal{G} = \bigcap_{h \in G} S(h)$, we take $[C] \in \bigcap_{h \in G} S(h)$ and we show that $[C] \geq [B]$. For contradiction, suppose $\mathsf{V}(C) \cap G \neq \emptyset$; then there exists some $h_1 \in \mathsf{V}(C) \cap G$. Then, since $[C] \in \bigcap S(h)$, $[C] \in S(h_1) = \{[A]: h_1 \notin \mathsf{V}(A)\}$, and so $h_1 \notin \mathsf{V}(C)$, which contradicts $h_1 \in \mathsf{V}(C) \cap G$. So necessarily $\mathsf{V}(C) \cap B \subseteq \mathsf{V}(C) \cap G = \emptyset$, i.e., $C \leq_0 B$, and so $[B] \leq [C]$. □

We can now finish the proof of Theorem 1.4. Let us denote the violator mapping of the concrete LP-problem $(\mathcal{B}/\!\sim, \leq, \mathcal{H})$ by $\widehat{\mathsf{V}}$. For $h \in H$, the fact $h \in \widehat{\mathsf{V}}(G)$ is by Lemma 3.7 equivalent to $w \circ \sigma^{-1}(G) < w \circ \sigma^{-1}(G \cup \{h\})$, i.e., $w(\sigma^{-1}(G)) < w(\sigma^{-1}(G) \cup \{S^{-1}(h)\})$. By Lemma 3.6 this happens if and only if $S^{-1}(h) \in \mathsf{V}(\sigma^{-1}(G))\}$, i.e., $h \in \sigma(\mathsf{V}(\sigma^{-1}(G)))$.

We have proved that the violators mapping of $(\mathcal{B}/\!\sim, \leq, \mathcal{H})$ is $\widehat{\mathsf{V}} = \sigma \circ \mathsf{V} \circ \sigma^{-1}$, i.e., V up to renaming of the elements of H. Thus Theorem 1.4 is proved. □

Example. Here we present some particular LP-type problems and we demonstrate the construction of their concrete representations.

Let a, b, c and d are the vertices of a unit square (in the anticlockwise order); let $H = \{a, b, c, d\}$. For $G \subseteq H$ let $w(G)$ be the radius of the smallest circle enclosing all the points of G (for $G = \emptyset$ put $w(G) = -\infty$). The corresponding violator space is described by the following table:

G	$\emptyset$	a	b	c	d	ab	ac	ad
$\mathsf{V}(G)$	$abcd$	bcd	acd	abd	abc	cd	$\emptyset$	bc

G	bc	bd	cd	abc	abd	acd	bcd	$abcd$
$\mathsf{V}(G)$	ad	$\emptyset$	ab	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

The bases are $\emptyset$, a, b, c, d, ab, ac, ad, bc, bd, cd; the only equivalent pair is $ac \sim bd$. There are no inconveniencies concerning differences between $\leq_0$ on sets and equivalence classes and $\leq_1$; the ordering $\leq_1$ is given by the Hasse diagram in Fig. 2. As the linear extension we may choose $\emptyset < a < b < c < d < ab < bc < cd < ad < [ac]$. Finally, the concrete representation S is as follows

G	a	b	c	d
$S(G)$	$a, ab, ad, [ac]$	$b, ab, bc, [ac]$	$c, bc, cd, [ac]$	$d, cd, ad, [ac]$

As the other example, consider the following LP problem in the positive orthant (rotated by 45 degrees for convenience). The constraints are the four halfplanes depicted in Fig. 3, the optimization direction is given by the arrow. Here the bases are $\emptyset$, a, b, c, d, ac, ad, bc, bd; the equivalence classes are $O = \emptyset$, $A = \{a\}$, $B = \{b\}$, $C = \{c\}$, $D = \{d\}$ and $Q = \{ac\} \sim \{ad\} \sim \{bc\} \sim \{bd\}$. Note that the equivalence classes correspond to the points in

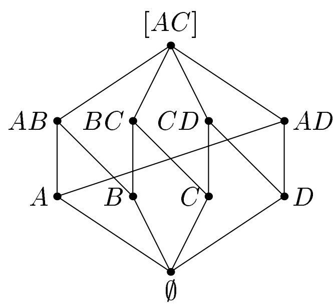

Figure 2: Hasse diagram of MinBall of the vertices of a square.

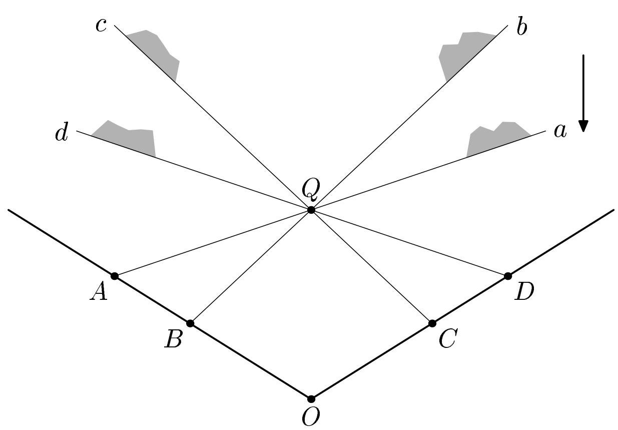

Figure 3: Illustration example—linear programming.

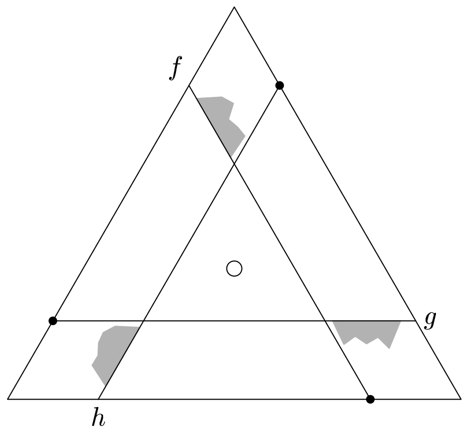

Figure 4: A cyclic violator space.

the plane. We have $O <_1 B <_1 A <_1 Q$ and $O <_1 C <_1 D <_1 Q$; we choose $<$ to be $O < B < A < C < D < Q$. The concrete representation is

G	a	b	c	d
$S(G)$	A, Q	A, B, Q	C, D, Q	D, Q

4 The role of acyclicity of violator spaces.

In this section we present an example of a cyclic violator space. However, we will see that all violator spaces arising from LP-type problems are acyclic.

A cyclic violator space. We begin with an intuitive geometric description; see Fig. 4. We consider a triangle without the center point. We say that a point is "locally smaller" if it is farther clockwise. The constraints in our violator space are the three halfplanes f, g, h. The locally smallest point within each halfplane is marked, and a halfplane violates a set of halfplanes if it does not contain the locally smallest point in their intersection.

Now we specify the corresponding violator space formally. We have $H = \{f, g, h\}$, and V is given by the following table:

G	$\emptyset$	f	g	h	f, g	f, h	g, h	f, g, h
$\mathsf{V}(G)$	f, g, h	h	f	g	h	g	f	$\emptyset$

This (H, V) is really a violator space, as we can easily check both consistency and locality. The bases are $\emptyset$, one-element sets, and H. We have $\{f\} \leq_0 \{h\} \leq_0 \{g\} \leq_0 \{f\}$, but none of the one-element bases are equivalent; i.e., $\leq_1$ is not antisymmetric.

Proposition 4.1 *Consider an LP-type problem $(H, w, W, \leq)$, and let V be the violators mapping. Then the violator space (H, V) is acyclic.*

Proof. First we observe that if for bases B, C we have $B \sim C$, then $\mathsf{V}(B) = \mathsf{V}(C)$ (Definition 3.4), and so $w(B) = w(C)$ (Lemma 2.4); i.e., the bases in the same equivalence class have the same value of w.

Now let $[B], [C]$ be two equivalence classes. If $[B] \leq_0 [C]$, then for some $B \in [B]$ and $C \in [C]$ we have $B \leq_0 C$; i.e., $B \cap \mathsf{V}(C) = \emptyset$. Using Lemma 2.2 and monotonicity we have $w(B) \leq w(B \cup C) = w(C)$. This shows that $[B] \leq_0 [C]$ implies $w(B) \leq w(C)$. If $[B] \leq_1 [C]$, then by chaining several $\leq_0$'s we get $w(B) \leq w(C)$ as well.

Furthermore, if $w(B) = w(C)$ (which is equal to $w(B \cup C)$), we use Lemma 2.4 to get $\mathsf{V}(B) = \mathsf{V}(C)$, i.e., $B \sim C$, so $[B] = [C]$. So $[B] \neq [C]$ implies $w(B) \neq w(C)$.

So, if $[B] <_1 [C]$, we have $w(B) < w(C)$. This proves that $\leq_1$ is necessarily antisymmetric (since $\leq$ is an ordering of W). □

5 Other results

Here we cite several additional results from [11].

The first result addresses the question of whether some values of V may be left unspecified while keeping the violator space uniquely determined.

Proposition 5.1 *Consider a set H, a subset $\mathcal{R} \subseteq 2^H$, and a mapping $\mathsf{V}\colon \mathcal{R} \to 2^H$ satisfying*

- *for all $G \subseteq H$ there exists $A_G \subseteq G$ such that $A_G \in \mathcal{R}$ and $\mathsf{V}(A_G) \cap G = \emptyset$,*
- *for all $A_1, A_2 \subseteq G \subseteq H$ such that $A_1, A_2 \in \mathcal{R}$ and $\mathsf{V}(A_i) \cap G = \emptyset$ for $i = 1, 2$, we have $\mathsf{V}(A_1) = \mathsf{V}(A_2)$,*

and consistency and locality, i.e.,

- *$G \cap \mathsf{V}(G) = \emptyset$ holds for all $G \in \mathcal{R}$, and*

- *for all $F \subseteq G \subseteq H$, where $F, G \in \mathcal{R}$ and $G \cap \mathsf{V}(F) = \emptyset$, we have $\mathsf{V}(G) = \mathsf{V}(F)$.*

Then there exists unique extension $\widehat{\mathsf{V}}$ of V to the whole 2^H such that $(H, \widehat{\mathsf{V}})$ is a violator space.

In particular, a violator space (H, V) is uniquely determined by H and the values of $\mathsf{V}(B)$ for bases.

The other results concern basic categorical properties of the violator spaces. The objects in the considered category are violator spaces. It remains to define what maps of violator spaces are *morphisms*. Although there may be several possibilities to consider, the mappings that preserve non-violation look the most natural as morphisms:

Definition 5.2 *Given violator spaces (G, V_G) and (H, V_H), we say that a mapping $\varphi: G \to H$ is a morphism if $x \notin \mathsf{V}_G(M) \Rightarrow \varphi(x) \notin \mathsf{V}_H\big(\varphi(M)\big)$, in other words $\varphi(x) \in \mathsf{V}_H\big(\varphi(M)\big) \Rightarrow x \in \mathsf{V}_G(M)$.*

It is immediate that the class of all violator spaces together with morphisms defined as above, composition of mappings, and identity mappings form a category. One might expect the following facts to hold, and indeed they are quite straightforward to prove:

Proposition 5.3 *A morphism $\varphi: (G, \mathsf{V}_G) \to (H, \mathsf{V}_H)$ is an epimorphism if and only if $\varphi: G \to H$ is surjective mapping.*

A morphism $\varphi: (G, \mathsf{V}_G) \to (H, \mathsf{V}_H)$ is a monomorphism if and only if $\varphi: G \to H$ is injective mapping.

A morphism $\varphi: (G, \mathsf{V}_G) \to (H, \mathsf{V}_H)$ is an isomorphism if and only if φ is a bijection of G and H and $\varphi(x) \in \mathsf{V}_H\big(\varphi(M)\big)$ exacly if $x \in \mathsf{V}_G(M)$.

The following proposition, with a rather lengthy proof, describes the categorical product of violator spaces.

Proposition 5.4 *Let I be a finite index set, and let (G_i, V_i) be violator spaces, $i \in I$. We put $H = \prod_{i \in I} G_i$ (the Cartesian product of the sets), and we let $\pi_i: H \to G_i$ be the projection onto the i-th component: $\pi_i\big((g_j)_{j \in I}\big) = g_i$. We define the mapping $\mathsf{V}_H: 2^H \to 2^H$ by*

$$\mathsf{V}_H(M) = \{(g_j)_{j \in I} : g_i \in \mathsf{V}_i(M_i) \text{ for some } i \in I\},$$

where $M_i = \pi_i(M) = \{g_i : (g_j)_{j \in I} \in M\}$.

Then (H, V_H) *is a violator space and together with mappings* π_i *it forms a product of the violator spaces* (G_i, V_i).

The product is acyclic if and only if all factors are acyclic.

We hope that products of violator spaces or other categorical operations might be a source of interesting examples for future research.

Acknowledgment

The first author would like to thank Nina Amenta for discussions concerning LP-type problems, possibly already forgotten by her as they took place many years ago, but nevertheless helpful for reaching the results in this paper. Both authors thank Bernd Gärtner for useful comments to the thesis [11].

References

[1] N. Amenta. Bounded boxes, Hausdorff distance, and a new proof of an interesting Helly theorem. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 340–347, 1994.

[2] N. Amenta. Helly-type theorems and generalized linear programming. *Discrete Comput. Geom.*, 12:241–261, 1994.

[3] N. Amenta. A short proof of an interesting Helly-type theorem. *Discrete Comput. Geom.*, 15:423–427, 1996.

[4] H. Björklund, S. Sandberg and S. Vorobyov. A discrete subexponential algorithm for parity games. In *Proc. 20th Sympos. Theoret. Aspects Comput. Sci.*, volume 2607 of *Lecture Notes Comput. Sci.*, pages 663–674. Springer-Verlag, 2003.

[5] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms*, 21:116–132, 1996.

[6] K. L. Clarkson. Las Vegas algorithms for linear and integer programming. *J. ACM*, 42:488–499, 1995.

[7] G. Kalai. A subexponential randomized simplex algorithm. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 475–482, 1992.

[8] J. Matoušek. On geometric optimization with few violated constraints. *Discrete Comput. Geom.*, 14:365–384, 1995.

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

[10] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Sympos. Theoret. Aspects Comput. Sci.*, volume 577 of *Lecture Notes Comput. Sci.*, pages 569–579. Springer-Verlag, 1992.

[11] P. Škovroň. Generalized linear programming. Masters Thesis, Charles University in Prague, Faculty of Mathematics and Physics, 2002. Available at `http://kam.mff.cuni.cz/~xofon/diplomka/`.