Clarkson's Algorithm for Violator Spaces

Yves Brise* Bernd Gärtner*

Abstract

Clarkson's algorithm is a two-staged randomized algorithm for solving linear programs, but it can also be applied to the more general LP-type problems which comprise a number of non-linear geometric problems. In 2006, it has been shown that the algorithm in its original form works for violator spaces too, which are a proper generalization of LP-type problems.

In this paper we show the following theoretical results: (*a*) It is shown, for the first time, that Clarkson's second stage can be simplified. (*b*) The previous simplifications of Clarkson's first stage carry over to the violator space setting. (*c*) Furthermore, we show the equivalence of violator spaces and partitions of the hypercube by hypercubes.

Keywords: Clarkson's Algorithm, Violator Space, LP-type Problem, Hypercube Partition

1 Introduction

Clarkson's randomized algorithm [2] is the earliest practical linear-time algorithm for linear programming with a fixed number of variables. Combined with a later algorithm by Matoušek, Sharir and Welzl [7], it yields the best (expected) worst-case bound in the unit cost model that is known today. The combined algorithm can solve any linear program with d variables and n constraints with an expected number of $O(d^2n + \exp(O(\sqrt{d \log d})))$ arithmetic operations [4]. Its generalization to *LP-type* problems makes it applicable to a number of non-linear and mostly geometric problems. For example, computing the smallest enclosing ball of n points in $\mathbb{R}^d$, or computing the distance between two d-polytopes with n facets.

Both stages of Clarkson's algorithm are based on random sampling and are conceptually very simple. The main idea is that we solve a subproblem subject to only a small number of (randomly chosen) constraints, but still have only few (of all) constraints that are violated by the solution of the subproblem. Some extra machinery was originally needed to make the analysis go through. For the first stage, it was already shown by Gärtner and Welzl that the extra machinery can be removed [5]. The result is what we call the *German Algorithm* below. In this paper, we do the removal also for the second stage, resulting in the *Swiss Algorithm*. We believe that the German and the Swiss Algorithm together represent the essence of Clarkson's approach.

Gärtner, Matoušek, Rüst, and Škovroň proved that Clarkson's original algorithm is applicable in a still broader setting: It actually works for the class of *violator spaces* [3]. At first glance, this seems to be yet another generalization to yet another abstract problem class, but it can be shown that it stops here: the class of violator spaces is the most general one for which Clarkson's algorithm still works [9]. It was unknown whether the analysis of the German Algorithm (the stripped-down version of Clarkson's first stage) also works for violator spaces. For LP-type problems, the analysis is nontrivial and constructs a "composite" LP-type problem. Here we show that this can still be done for violator spaces, in essentially the same way.

For the Swiss Algorithm (the stripped-down version of Clarkson's second stage), we provide the first analysis at all.

See [1] for some additional notes on violator spaces.

The German Algorithm (`GA`). Let us stick to the problem of finding the smallest enclosing ball of a set of n points in $\mathbb{R}^d$. The algorithm proceeds in rounds and maintains a working set G, initialized with a subset R of r points drawn at random. In each round, the smallest enclosing ball of G is being computed (by some other algorithm). For the next round, the points that lie outside this ball are added to G. The algorithm terminates as soon as everybody is happy with the smallest enclosing ball of G. A point is happy, of course, if it lies inside a ball.

The crucial fact is this: the number of rounds is at most $d+2$, and for $r \approx d\sqrt{n}$, the expected maximum size of G is bounded by $O(d\sqrt{n})$. This means that `GA` reduces a problem of size n to $d+2$ problems of expected size $O(d\sqrt{n})$. We call this the German Algorithm, because it takes – typically German – one decision in the beginning which is then efficiently pulled through.

The Swiss Algorithm (`SA`). Like `GA`, this algorithm proceeds in rounds, but it maintains a voting box that initially contains one slip per point. In each round, a set of r slips is drawn at random from the voting box,

*Department of Computer Science, Swiss Federal Institute of Techology (ETHZ), ybrise | gaertner@inf.ethz.ch

and the smallest enclosing ball of the corresponding set R is computed (by some other algorithm). For the next round, the number of slips of the unhappy points is doubled. The algorithm terminates as soon as everybody is happy with the smallest enclosing ball of the sample R.

Below, we will prove the following: if $r \approx d^2$, the expected number of rounds is $O(\log n)$. This means that SA reduces a problem of size n to $O(\log n)$ problems of size $O(d^2)$. We call this the Swiss Algorithm, because it takes – typically Swiss – many independent local decisions that magically fit together in the end.

Hypercube partitions. A hypercube partition is a partition of the vertices of the hypercube such that every element of the partition is the set of vertices of some subcube. It was known that every *nondegenerate* violator space induces a hypercube partition [8, 6]. We prove here that also the converse is true, meaning that we obtain an alternative characterization of the class of violator spaces.

2 Prerequisites

2.1 The Sampling Lemma

The following lemma is due to Gärtner and Welzl in [5] and was adapted to violator spaces in [3]. Let S be a set of size n, and $\varphi : 2^S \to \mathbb{R}$ a function that maps any set $R \subseteq S$ to some value $\varphi(R)$. Define

$$\mathsf{V}(R) := \{s \in S \backslash R \,|\, \varphi(R \cup \{s\}) \neq \varphi(R)\}, \quad (1)$$

$$\mathsf{X}(R) := \{s \in R \,|\, \varphi(R \backslash \{s\}) \neq \varphi(R)\}. \quad (2)$$

$\mathsf{V}(R)$ is the set of *violators* of R, while $\mathsf{X}(R)$ is the set of *extreme elements* in R. Obviously,

$$s \text{ violates } R \Leftrightarrow s \text{ is extreme in } R \cup \{s\}. \quad (3)$$

For a random sample R of size r, i.e., a set R chosen uniformly at random from the set $\binom{S}{r}$ of all r-element subsets of S, we define random variables $\mathsf{V}_r : R \mapsto |\mathsf{V}(R)|$ and $\mathsf{X}_r : R \mapsto |\mathsf{X}(R)|$, and we consider the expected values $v_r := \mathrm{E}[\mathsf{V}_r]$, and $x_r := \mathrm{E}[\mathsf{X}_r]$.

Lemma 1 (Sampling Lemma, [3]) *For $0 \leq r < n$, $v_r/(n-r) = x_{r+1}/(r+1)$.*

Proof. See the full version of this paper [1]. □

2.2 Violator Spaces

Definition 2 *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 the following two conditions are fulfilled.*

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

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

Condition (i) is usually known as consistency*, and (ii) is referred to as* locality*.*

Lemma 3 ([3]) *Any violator space (H, V) satisfies* monotonicity *defined as follows:*

$\mathsf{V}(F) = \mathsf{V}(G)$ implies $\mathsf{V}(E) = \mathsf{V}(F) = \mathsf{V}(G)$ for all sets $F \subseteq E \subseteq G \subseteq H$.

Proof. Assume $\mathsf{V}(E) \neq \mathsf{V}(F), \mathsf{V}(G)$. Then locality yields $\emptyset \neq E \cap \mathsf{V}(F) = E \cap \mathsf{V}(G)$ which contradicts consistency. □

Definition 4 *Consider a violator space (H, V).*

(i) *We say that $B \subseteq H$ is a* basis *if for all proper subsets $F \subset B$ we have $B \cap \mathsf{V}(F) \neq \emptyset$. For $G \subseteq H$, a basis of G is a minimal subset B of G with $\mathsf{V}(B) = \mathsf{V}(G)$. A basis in (H, V) is a basis of some set $G \subseteq H$.*

(ii) *The* combinatorial dimension *of (H, V), denoted by $\dim(H, \mathsf{V})$, is the size of the largest basis in (H, V).*

(iii) *(H, V) is* nondegenerate *if every set set $G \subseteq H$, $|G| \geq \dim(H, \mathsf{V})$, has a unique basis. Otherwise (H, V) is* degenerate*.*

Corollary 5 (of Lemma 1) *Let (H, V) be a violator space of combinatorial dimension d, and $|H| = n$. If we choose a subset $R \subseteq H$, $|R| = r \leq n$, uniformly at random, then $\mathrm{E}[|\mathsf{V}(R)|] \leq d\frac{n-r}{r+1}$.*

Proof. The corollary follows from Lemma 1, with the observation that $|X(R)| \leq d$, $\forall R \subseteq H$. □

3 Clarkson's Algorithm Revisited

Clarkson's algorithm computes a basis of some violator space (H, V), $n = |H|$. It consists of two separate stages and the Brute Force Algorithm (BFA). The results about the running time and the size of the sets involved are summarized in Theorem 6 and Theorem 11.

The main idea of both stages (GA and SA) is the following: We draw a random sample $R \subseteq H$ of size $r = |R|$ and then compute a basis of R using some other algorithm. The crucial point here is that $r \ll n$. Obviously, such an approach may fail to find a basis of H, and we might have to enter a second round. That is the point at which GA and SA most significantly differ.

In both stages we assume that the size of the ground set, i.e., n, is larger than r, such that we can actually draw a sample of that size.

3.1 The German Algorithm (GA)

This algorithm works as follows. Let (H, V) be a violator space, $|H| = n$, and $\dim(H, \mathsf{V}) = d$. We draw a random sample $R \subseteq H$, $r = d\sqrt{n/2}$, only once, and initialize our working set G with R. Then we enter a repeat loop, in which we compute a basis B of G (by

calling the Swiss Algorithm) and find all violators of B in H. If there are none, we are done and return B. Otherwise, we add those violators to our working set G and repeat the procedure.

The analysis shows that (i) the number of rounds is bounded by $d+1$, and (ii) the size of G in any round is bounded by $O(d\sqrt{n})$. Note that $\mathsf{V}|_G$ denotes the restriction of the violator mapping to the set G.

Algorithm 1: $\texttt{GA}(H, \mathsf{V})$

input : Violator space (H, V), $|H| = n$, and $\dim(H, \mathsf{V}) = d$
output: A basis B of (H, V)

$r \leftarrow d\sqrt{n/2}$;
Choose R with $|R| = r$, $R \subseteq H$ u.a.r.;
$G \leftarrow R$;
repeat
  $B \leftarrow \texttt{SA}(G, \mathsf{V}|_G)$;
  $G \leftarrow G \cup \mathsf{V}(B)$;
until $\mathsf{V}(B) = \emptyset$;
return B

Theorem 6 *Let (H, V) be a violator space of combinatorial dimension d, and $n = |H|$. Then the algorithm GA computes a basis of (H, V) with at most $d+1$ calls to SA, with an expected number of at most $O(d\sqrt{n})$ constraints each.*

Proof. The proof is a modification of a similar proof found in [5], where it is employed in the framework of LP-type problems; see [1]. □

3.2 The Swiss Algorithm (SA)

Let the input be a violator space (H, V), $|H| = n$, and $\dim(H, \mathsf{V}) = d$. We will compute a basis B of H.

First, let us introduce the notation $R^{(i)}$, $B^{(i)}$, and $V^{(i)}$ for $i \geq 1$ for the sets R, B and $\mathsf{V}(R)$ in round i of SA respectively. The set $B^{(i)}$ is a basis of $R^{(i)}$ and $V^{(i)} = \mathsf{V}(R^{(i)}) = \mathsf{V}(B^{(i)})$.

After the initialization we enter the first round and choose the random sample $R^{(1)}$ of size $r = 2d^2$ uniformly at random from H. Then we compute an intermediate basis $B^{(1)}$ of the violator space $(R^{(1)}, \mathsf{V}|_{R^{(1)}})$ by using BFA as a black box. In the next step we compute the set of violated constraints, i.e., $V^{(1)}$. So far, it is the same thing as in GA. But now, instead of enforcing the violated constraints, we just increase their probability that they will be chosen in the next round. Repeat this procedure until the solution to the subproblem has no more violators.

To formalize the above, we associate the *multiplicity* $\mu_h \in \mathbb{N}$ with every $h \in H$. For an arbitrary set $F \subseteq H$ we define the *cumulative multiplicity* as $\mu(F) := \sum_{h \in F} \mu_h$. For $i \geq 0$ we will use $\mu_h^{(i)}$ (and $\mu^{(i)}(F)$) to denote the (cumulative) multiplicity at the end of round i. We define $\mu_h^{(0)} := 1$ for any $h \in H$, and therefore $\mu^{(0)}(F) = |F|$.

To increase the probability that a constraint $h \in V^{(i)}$ is chosen in the random sample of round $i+1$ we double the multiplicity of h, i.e., $\mu_h^{(i)} = 2\mu_h^{(i-1)}$. The random sample is chosen as follows. We consider a collection of elements in which every constraint h appears with its multiplicity $\mu_h^{(i)}$. From this we draw a r-element subset u.a.r. and then discard multiple entries. In general, our random sample R therefore has size $1 \leq |R| \leq r$.

Algorithm 2: $\texttt{SA}(H, \mathsf{V})$

input : Violator space (H, V), $|H| = n$, and $\dim(H, \mathsf{V}) = d$
output: A basis B of (H, V)

$\mu_h \leftarrow 1$ for all $h \in H$;
$r \leftarrow 2d^2$;
repeat
  choose random R from H according to μ;
  $B \leftarrow \texttt{BFA}(R, \mathsf{V}|_R)$;
  $\mu_h \leftarrow 2\mu_h$ for all $h \in \mathsf{V}(B)$;
until $V(B) = \emptyset$;
return B

The following lemma guarantees that in every round the multiplicity of at least one element of the final basis is doubled.

Lemma 7 (Observation 22, [3]) *Let (H, V) be a violator space, $F \subseteq G \subseteq H$, and $G \cap \mathsf{V}(F) \neq \emptyset$. Then $G \cap \mathsf{V}(F)$ contains at least one element from every basis of G.*

The analysis of SA will show that the elements in any basis B of H will increase their multiplicity so quickly that they are chosen with high probability after a logarithmic number of rounds. This, of course, means that the algorithm will terminate, because there will be no violators. Formally, we will consider a modification of SA that runs forever, regardless of the current set of violators. Let us call the modified algorithm SA_forever. We call a particular round i *controversial* if $V^{(i)} \neq \emptyset$. Furthermore, let C_ℓ be the event that the first ℓ rounds are controversial in SA_forever.

Lemma 8 *Let (H, V) be a violator space, $|H| = n$, $\dim(H, \mathsf{V}) = d$, B any basis of H, and $k \in \mathbb{N}$ some positive integer. Then, in SA_forever, the following holds for the expected cumulative multiplicity of B after kd rounds, $2^k \Pr[C_{kd}] \leq \mathrm{E}[\mu^{(kd)}(B)]$.*

Proof. In any controversial round, Lemma 7 asserts that $B \cap V^{(i)} \neq \emptyset$. So, in every controversial round,

the multiplicity of at least one element in B is doubled. Therefore, by conditioning on the event that the first kd rounds are controversial, there must be a constraint in B that has been doubled at least k times (recall that $|B| \leq d$). It follows that $\mathrm{E}[\mu^{(kd)}(B)] = \mathrm{E}[\mu^{(kd)}(B) \,|\, C_{kd}] \Pr[C_{kd}] + \mathrm{E}[\mu^{(kd)}(B) \,|\, \overline{C_{kd}}] \Pr[\overline{C_{kd}}] \geq 2^k \Pr[C_{kd}]$. □

Lemma 9 *Let (H, V) be a violator space, $|H| = n$, $\dim(H, \mathsf{V}) = d$, B any basis of H, and $k \in \mathbb{N}$ some positive integer. Then, in* `SA_forever`*, the following holds for the expected cumulative multiplicity of B after kd rounds, $\mathrm{E}[\mu^{(kd)}(B)] \leq n\left(1 + \frac{d}{r}\right)^{kd}$.*

Proof. The detailed proof can be found in [1]. What we do is basically construct a different violator space in every round, that captures the sampling process, and then we apply the sampling lemma to this new space. This lets us bound from above the cumulative multiplicity of the original space after some number of rounds. □

Using $\ell = kd$, and combining Lemmata 8 and 9, we now know that $2^k \Pr[C_\ell] \leq n\left(1 + \frac{d}{r}\right)^\ell$, which gives us a useful upper bound on $\Pr[C_\ell]$, because the left-hand side power grows faster than the right-hand side power as a function of ℓ, given that r is chosen large enough.

Let us choose $r = c\,d^2$ for some constant $c > \log_2 e \approx 1.44$. We obtain $\Pr[C_\ell] \leq n\left(1 + \frac{1}{c\,d}\right)^\ell / 2^k \leq n\,2^{(\ell \log_2 e)/(c\,d) - k}$, using $1 + x \leq e^x = 2^{x \log_2 e}$ for all x. This further gives us

$$\Pr[C_\ell] \leq n\alpha^\ell, \tag{4}$$

where $\alpha = \alpha(d, c) = 2^{(\log_2 e - c)/(c\,d)} < 1$. This implies the following tail estimate.

Lemma 10 *For any $\beta > 1$, the probability that* `SA_forever` *starts with at least $\lceil \beta \log_{1/\alpha} n \rceil$ controversial rounds is at most $n^{1-\beta}$.*

Proof. The probability for at least this many leading controversial rounds is at most $\Pr[C_{\lceil \beta \log_{1/\alpha} n \rceil}] \leq n\alpha^{\lceil \beta \log_{1/\alpha} n \rceil} \leq n\alpha^{\beta \log_{1/\alpha} n} = nn^{-\beta} = n^{1-\beta}$. □

So, we can bound the expected number of leading controversial rounds in `SA_forever`, and this bounds the expected number of rounds in `SA`, because `SA` terminates upon the first non-controversial round it encounters.

Theorem 11 *Let (H, V) be a violator space, $|H| = n$, and $\dim(H, \mathsf{V}) = d$. Then the algorithm* `SA` *computes a basis of H with an expected number of at most $O(d \ln n)$ calls to* `BFA`*, with at most $O(d^2)$ constraints each.*

Proof. A detailed computation of the expected number of leading controversial rounds can be found in [1]. The second part of the theorem follows immediately, because we make calls to `BFA` with at most cd^2 constraints each. □

4 Hypercube Partitions

Let H be a finite set. Consider the graph on the vertices 2^H, where two vertices F, G are connected by an edge if they differ in exactly one element, i.e., $G = F \dot\cup \{h\}$, $h \in H$. This graph is a hypercube of dimension $n = |H|$. For the sets $A \subseteq B \subseteq H$, we define $[A, B] := \{C \subseteq H \mid A \subseteq C \subseteq B\}$ and call any such $[A, B]$ an *interval*. A *hypercube partition* is a partition $\mathcal{P}$ of 2^H into (disjoint) intevals.

Let (H, V) be a violator space. We call two sets $F, G \subseteq H$ *equivalent* if $\mathsf{V}(F) = \mathsf{V}(G)$, and let $\mathcal{H}$ be the partition of 2^H into equivalence classes w.r.t. this relation. We call $\mathcal{H}$ the *violation pattern* of (H, V).

Theorem 12 *Any hypercube partition $\mathcal{P}$ is the violation pattern of some nondegenerate violator space (H, V).*

Proof. The proof (including some auxiliary Lemmata) can be found in the full version of this paper [1]. □

5 Conclusion

We analyzed Clarkson's algorithm in what we believe to be its most general as well as natural setting. Additionally, we have given an equivalence between nondegenerate violator spaces and hypercube partitions, which could help identifying further applications in computational geometry as well as other fields of computer science. A major challenge will be to establish a subexponential analysis for the framework of violator spaces (as there already exists for LP's and LP-type problems).

References

[1] Yves Brise and Bernd Gärtner. Clarkson's algorithm for violator spaces, arXiv:0906.4706v2, 2009.

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

[3] Bernd Gärtner, Jiří Matoušek, Leo Rüst, and Petr Škovroň. Violator spaces: Structure and algorithms. *Discrete Applied Mathematics*, 156(11):2124–2141, 2008.

[4] Bernd Gärtner and Emo Welzl. Linear programming - randomization and abstract frameworks. In *Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1046 of *Lecture Notes in Computer Science*, pages 669–687. Springer, 1996.

[5] Bernd Gärtner and Emo Welzl. A Simple Sampling Lemma: Analysis and Applications in Geometric Optimization. *Discrete & Computational Geometry*, 25(4):569–590, 2001.

[6] Jiří Matoušek. Removing degeneracy in LP-type problems revisited. *Discrete & Computational Geometry*, 2008.

[7] Jiří Matoušek, Micha Sharir, and Emo Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.

[8] Jiří Matoušek and Petr Škovroň. Removing degeneracy may require a large dimension increase. *Theory of Computing*, 3(1):159–177, 2007.

[9] Petr Škovroň. *Abstract Models of Optimization Problems*. PhD thesis, Charles University, Prague, 2007.