Computational Aspects of Helly's Theorem and its Relatives

David Avis* and Michael E. Houle**

*School of Computer Science, McGill University
Montréal, Québec, Canada H3A 2A7
**Department of Information Science, University of Tokyo
Bunkyo-ku, Tokyo 113, Japan

(extended abstract)

1 Introduction

Of the known results in the field of convexity theory, perhaps the most famous is the existence theorem due to Helly which relates the intersection of a collection of convex sets with the intersection of its subcollections [2]:

Theorem 1 (Helly) *Let $C = \{C_1, C_2, \ldots, C_n\}$ be a collection of convex sets in $\Re^d$. Then if every $d+1$ sets intersect, there exists a point common to all the sets.*

This result has had numerous applications in proving other combinatorial statements, in particular those of the general form: *If a certain type of collection is such that each of its subfamilies of k members has a certain property, then the whole collection has that property* (see [1] for a partial survey). An algorithmic implication of such statements is that to determine whether the collection has the property, it suffices to determine whether every subfamily of k members has the property. In the case of Helly's theorem itself, if we had an algorithm which determined whether any $d+1$ members of C have a common intersection, then by applying this algorithm to all $\binom{n}{d+1}$ subfamilies of $d+1$ members, we could determine whether all sets of C intersect. However, no *constructive* applications had been discovered for Helly's theorem or its relatives.

In the next section, we outline a general method by which a point in the intersection of n convex sets in $\Re^d$ may be computed, assuming the existence of an algorithm for finding a point in the intersection of any $d+1$ of these sets. The algorithm will be seen to require time proportional to $\binom{n}{d+1} \cdot f(d+1)$, where $f(d+1)$ is the time required to find a point in the intersection of $d+1$ sets. Although this method perhaps cannot compete against fast algorithms for special sets such as half-spaces, it may be efficient in cases where the sets are very complex.

2 Computing an Intersection Point Using Radon Partitions

A *Radon partition* of a set of n points in $\Re^d$ is a partition into two sets S and T such that their convex hulls $CH(S)$ and $CH(T)$ intersect. Radon's theorem [5] guarantees the existence of a Radon partition whenever n is at least $d+2$. From this, the following lemma can be proved:

Lemma 2 *Let* $\{C_1, C_2, \ldots, C_{d+2}\}$ *be a collection of convex sets in* $\Re^d$, *and let* $P = \{p_1, p_2, \ldots, p_{d+2}\}$ *be a set of points such that* $p_i \in \bigcap_{j \neq i} C_j$ *for* $0 \leq i \leq d+2$. *If* S *and* T *form a Radon partition of* P, *then* $CH(S) \cap CH(T) \subseteq \bigcap_j C_j$.

This lemma may be used to prove Helly's theorem by induction. Indeed, Radon first used these arguments to this end in 1921 [5]. Implicit in the lemma, moreover, is the basis of a recursive algorithm for finding the common intersection of convex sets.

Given a collection $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ of convex sets in $\Re^d$, let us assume that we have available a "black box" or "oracle" which accepts as input the indices of $d+1$ sets of $\mathcal{C}$, and which gives as its output a point in the intersection of these sets, or reports that the sets do not intersect, as the case may be. We shall also assume that a call to the oracle and the computation of a Radon partition both take unit time, since these operations are independent of the size of $\mathcal{C}$.

For each $k = 1, 2, \ldots, n-d$, we will compute an intersection point for the collection

$$\{C_1, C_2, \ldots, C_k, C_{x_1}, C_{x_2}, \ldots, C_{x_d}\}, \quad \{x_1, x_2, \ldots, x_d\} \subseteq \{k+1, \ldots, n\}. \tag{1}$$

Note that when $k = n-d$, this is the required intersection point. If at any time a call to the oracle reveals that some subcollection has no intersection, then the algorithm terminates.

For $k = 1$, an intersection point of each of the $\binom{n-1}{d}$ families $\{C_1, C_{x_1}, C_{x_2}, \ldots, C_{x_d}\}$ may be obtained directly from the oracle. In general, having found all of the intersection points for the families in (1) up to $k-1 \geq 1$, we compute the values for k by taking a Radon partition of the intersection points of the following $d+2$ families:

$$\begin{aligned} &\mathcal{K} \setminus C_k, \\ &\mathcal{K} \setminus C_{x_i}, \quad \text{for } 1 \leq i \leq d, \quad \text{and} \\ &\{C_k, C_{x_1}, C_{x_2}, \ldots, C_{x_d}\}, \end{aligned}$$

where

$$\mathcal{K} = \bigcap_{j=1}^{k} C_j \cap \bigcap_{i=1}^{d} C_{x_i}.$$

From Lemma 2, it follows that the intersection point of the hulls of the Radon partition is contained in $\mathcal{K}$. Therefore, to compute a point in the sets of (1) for each value of $k \geq 2$ involves finding $\binom{n-k}{d}$ Radon partitions and $\binom{n-k}{d}$ calls to the oracle. The total number of Radon partitions computed is

$$\sum_{k=2}^{n-d} \binom{n-k}{d} = \sum_{i=d}^{n-2} \binom{i}{d} = \binom{n-1}{d+1}.$$

Taking the case $k = 1$ into consideration, the number of calls to the oracle is thus

$$\binom{n-1}{d} + \binom{n-1}{d+1} = \binom{n}{d+1},$$

one for each subfamily of C consisting of $d+1$ members.

The algorithm just described is optimal in the number of calls to the oracle, since we can construct families of n convex sets for which all but one of its subfamilies of $d+1$ members has a non-empty intersection. However, it is certainly not optimal in terms of space, since at iteration $k = k'$ we require that the results of $k = k' - 1$ be kept available.

3 Applications

Just as many combinatorial results follow from Helly's theorem, the method presented in the previous section has many applications. In this concluding section, we will mention but three, although we will not justify our claims in this extended abstract. In the full paper, other extensions will also be discussed.

The first algorithm follows from an application of Helly's theorem due to Vincensini [6] and Klee [4]:

Theorem 3 *Let $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ be a collection of convex sets in $\Re^d$, and let C be another convex set. Given an algorithm that accepts as input $d+1$ sets of $\mathcal{C}$, and returns a translate of C contained in the intersection of these sets (or reports its non-existence), then a translate of C contained in the intersection of the entire collection may be found (or its non-existence reported) in time proportional to $\binom{n}{d+1} \cdot f(d+1)$. Here, $f(d+1)$ is the time taken by the algorithm.*

Next, let $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ be a family of n sets in $\Re^d$, some labeled *red* and the others labeled *green*. A *linear separator* of $\mathcal{S}$ is a hyperplane h such that all the red sets are contained in one open half-space bounded by h, and all the green sets are contained in the other. A *spherical* *separator* of S is a hypersphere s (possibly degenerate) such that all the red sets are contained in one connected region of $\Re^d \setminus s$, and all the green sets are contained in the other.

Recently, Houle [3] showed a reduction of the problem of determining whether n sets can be linearly separated to the problem of determining whether n convex sets of the same dimension have a common intersection point. In the case of spherical separation, the n convex sets are of one higher dimension. Using these transformations, we arrive at the following extensions of the intersection algorithm:

Theorem 4 *Given an algorithm that accepts as input $d+2$ sets of S, and returns a linear separator of this subset (or reports its non-existence), a linear separator for the entire collection may be found (or its non-existence reported) in time proportional to $\binom{n}{d+1} \cdot f(d+2)$. Here, $f(d+2)$ is the time taken by the algorithm.*

Theorem 5 *Given an algorithm that accepts as input $d+3$ sets of S, and returns a spherical separator of this subset (or reports its non-existence), a spherical separator for the entire collection may be found (or its non-existence reported) in time proportional to $\binom{n}{d+2} \cdot f(d+3)$, Here, $f(d+3)$ is the time taken by the algorithm.*

References

[1] Danzer, L., Grünbaum, B. and Klee, V. Helly's Theorem and its Relatives, in *Convexity, AMS Proc. Symp. Pure Math.* 7, 1963, pp. 101-181.

[2] Helly, E. Über Mengen konvexer Körper mit gemeinschaftlichen Punkten, *Jber. Deutsch. Math.-Verein.* 32, 1923, pp. 175-176.

[3] Houle, M. E. Theorems on the Existence of Separating Surfaces, *Discrete Comput. Geom.* 6, 1991, pp. 49-56.

[4] Klee, V. The Critical Set of a Convex Body, *Amer. J. Math.* 75, 1953, 178-188.

[5] Radon, J. Mengen konvexer Körper, die einen gemeinsamen Punkt enthalten, *Math. Ann.* 83, 1921, pp. 113-115.

[6] Vincensini, P. Sur une extension d'un théorème de M. J. Radon sur les ensembles de corps convexes, *Bull. Soc. Math. France* 67, 1939, pp. 115-119.