Decomposition of polytopes and polynomials*

Shuhong Gao[1] and Alan Lauder[2]

[1] Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-0975, USA. E-mail: sgao@math.clemson.edu.

[2] Mathematical Institute, Oxford University, Oxford OX1 3LB, U.K. E-mail: lauder@maths.ox.ac.uk.

Abstract. Motivated by a connection with the factorization of multivariable polynomials, we study integral convex polytopes and their integral decompositions in the sense of the Minkowski sum. We first show that deciding decomposability of integral polygons is NP-complete then present a pseudo-polynomial time algorithm for decomposing polygons. For higher dimensional polytopes, we give a heuristic algorithm which is based upon projections and uses randomization. Applications of our algorithms include absolute irreducibility testing and factorization of polynomials via their Newton polytopes.

1 Introduction

It is well-known that the theory of convex polytopes has many applications across mathematics and computer science [2,9,12,28]. One such application is to polynomial factorization, and motivated by this connection we discuss decomposition algorithms for polytopes. Given a multivariable polynomial one may associate with it, in a way we shall fully explain in Section 2, an integral polytope called its Newton polytope. It was observed by Ostrowski in 1921 that if the polynomial factors then its Newton polytope decomposes, in the sense of the Minkowski sum, into the Newton polytopes of the factors. The ramifications of this simple observation are two-fold. Firstly, criteria which ensure polytope indecomposability can be used to construct families of irreducible, indeed absolutely irreducible, polynomials. Secondly, algorithms which test whether a polytope is decomposable and construct decompositions may be useful in factoring polynomials. Of course, such criteria and algorithms are also of independent interest and may have other applications. Indecomposability conditions were explored by the first author in [4] and will be discussed further in Section 3. Our main focus will be, however, on the second application, that is on algorithms for decomposing polytopes.

We first show that the problem of testing whether a polytope is indecomposable is NP-complete even in dimension two, so there does not exist, unless NP = P, a genuinely efficient algorithm for decomposing polytopes. However,

* The first author was supported in part by NSF under Grant #DMS9970637 and NSA under Grant #MDA904-00-1-0048. The second author gratefully acknowledges the support of the Marr Educational Trust and Wolfson College, Oxford, and thanks Dominic Welsh for his help and encouragement.

we present a "pseudo-polynomial" time algorithm (see [6]) for testing indecomposability in dimension two and a modified version which also allows one to count the number of decompositions and find summands. We also discuss a heuristic algorithm which uses randomization for testing higher dimensional polytopes for indecomposability. In Section 5, we describe applications of our algorithms to polynomials with respect to their irreducibility and factorization. In particular, we touch upon an open problem in polynomial factorization which we now describe. In his survey paper on polynomial factorization [16], Kaltofen concludes with several open problems one of which, due to B. Sturmfels, is stated as follows: "From the support vectors $(e_{j1}, \dots, e_{jn})$ of a sparse polynomial $\sum_{j=1}^{t} a_{e_{j1},\dots,e_{jn}} X_1^{e_{j1}} \cdots X_n^{e_{jn}}$, compute by geometric considerations the support vectors of all possible factorizations". This problem can be attacked by our polytope method, although it must be noted that we are unable to give a complete solution. The basic idea runs as follows: Given a bivariable polynomial, we can compute its Newton polytope and then find all the integral summands of this polytope. The summands correspond to the Newton polytopes of all the possible factors of the polynomial. The integral points in a summand give the support vectors of the factor corresponding to the summand.

The remainder of the paper is organized in the following way. Section 2 contains the necessary background material on the theory of convex polytopes and in Section 3 we discuss some preliminary results on polytope indecomposability which shall be useful to us but are also of independent interest. Section 4 is devoted to algorithms and is further divided into two parts: In Section 4.1 we present algorithms for both testing polygons for decomposability and counting and constructing decompositions of polygons. Section 4.2 contains a heuristic randomized algorithm for higher dimensional polytopes based upon projections down to dimension two. Finally, in Section 5 we discuss applications of these algorithms to absolute irreducibility testing and polynomial factorization.

2 Polynomials and Newton polytopes

2.1 Background geometry and algebra

Before describing the connection between polynomials and polytopes, we recall some terminology and results from the theory of convex polytopes ([13]). Let $\mathbb{R}$ denote the field of real numbers and $\mathbb{R}^n$ the Euclidean n-space. A *convex set* in $\mathbb{R}^n$ is a set such that the points on the line segment joining any two points of the set lie in the set; the *convex hull* of a set of points is the smallest convex set which contains them; and the convex hull of a finite set of points is called a *convex polytope*. A point of a polytope is called a *vertex* (or *extreme point*) if it does not belong to the interior of any line segment contained in the polytope. A polytope is always the convex hull of its vertices. A hyperplane *cuts* a polytope if both of the open half spaces determined by it contain points of the polytope. A hyperplane which does not cut a polytope, but has a non-empty intersection with it is called a *supporting hyperplane*. The intersection of a supporting hyperplane and a polytope is a *(proper) face*, and the union of all (proper) faces is the *boundary*. One may equivalently define a vertex to be a 0-dimensional face, and 1-dimensional faces are known as *edges*.

For two subsets A and B in $\mathbb{R}^n$, define their *Minkowski sum* to be $A+B = \{a+b \mid a \in A, b \in B\}$. We call A and B the *summands* of $A+B$. It is easy to show that the Minkowski sum of two convex polytopes is a convex polytope.

Let $f \in K[X_1, \ldots, X_n]$ be a nonconstant polynomial where K is an arbitrary field. We call f *absolutely irreducible* over K if it has no non-trivial factors over the algebraic closure of K. Suppose

$$f = \sum a_{i_1 \ldots i_n} X_1^{i_1} \cdots X_n^{i_n}.$$

For each term with $a_{i_1 \ldots i_n} \neq 0$, the corresponding exponent vector $(i_1, \ldots, i_n)$, viewed in $\mathbb{R}^n$, is called a support vector of f. Define $Supp(f)$ to be the set of all support vectors of f, i.e.,

$$Supp(f) = \{(i_1, \ldots, i_n) \mid a_{i_1 \ldots i_n} \neq 0\}.$$

Note that $Supp(f)$ is empty if $f = 0$. The *total degree* of f, where $f \neq 0$, is the maximum value of $\sum_{1 \leq j \leq n} i_j$ over all $(i_1, \ldots, i_n) \in Supp(f)$. The convex hull of the set $Supp(f)$, denoted P_f, is known as the *Newton polytope* of f.

The following lemma was observed by Ostrowski [21] in 1921 (see also [22, Theorem VI, p. 226]).

Lemma 1. *Let $f, g, h \in K[X_1, \ldots, X_n]$ with $f = gh$. Then $P_f = P_g + P_h$.*

An *integral polytope* is a polytope whose vertices have integer coordinates, and we say that an integral polytope is *integrally decomposable*, or simply *decomposable*, if it can be written as a Minkowski sum of two integral polytopes, each of which has more than one point. A summand in an integral decomposition is called an *integral summand*. We say an integral polytope is *integrally indecomposable*, or simply *indecomposable*, if it is not decomposable. The Newton polytope of a polynomial is certainly integral and if the polynomial factors into two polynomials each of which has at least two terms, then by Lemma 1 its Newton polytope must be decomposable. Thus we have the following simple irreducibility criterion from [4].

Corollary 2 (Irreducibility Criterion). *Let $f \in K[X_1, \ldots, X_n]$ with f not divisible by any X_i for $1 \leq i \leq n$. If the Newton polytope of f is integrally indecomposable, then f is absolutely irreducible.*

In Section 3, we shall discuss in more detail constructions of indecomposable polytopes and show how to get indecomposable polytopes of high dimension from those of lower dimensions. From these indecomposable polytopes one can easily give explicitly many infinite families of polynomials which are absolutely irreducible when considered over any field.

2.2 Relevant computational problems

From a computational point of view, the following problem is of interest.

Problem 3. Given an integral polytope, say as its list of vertices, decide whether it is integrally indecomposable.

This problem is not only pertinent to the study of polynomial factorization, but is a natural problem to consider and as such may be useful in other applications. Here the input size is the length of the binary representation of the coordinates of the vertices. Note that in our applications the polytope will be presented as the convex hull of a set of integral points. There is a large literature on computing the convex hull of any finite set of points in $\mathbb{R}^n$; see [9, pages 361–375]. In particular, the convex hull of t points in a plane can be computed in time $O(t \log t)$ [10]. Any of these algorithms can be used to compute the vertices of the Newton polytope of a given polynomial and we shall ignore this computational problem in the presentation of our algorithms.

As mentioned before, the above problem is NP-complete, thus we shall be contented with algorithms that are "efficient" in terms of some more generous measure, say the volume of polytopes. In Section 4 we give such an algorithm for polytopes in $\mathbb{R}^2$ and we also present a heuristic algorithm for higher dimensional polytopes which uses randomization. It is an open problem to develop an "efficient" deterministic or even randomized algorithm for testing general integral polytopes for indecomposability.

For a decomposable integral polytope, it is desirable to find all of its integral summands. Here we should identify polytopes that are translations of each other.

Problem 4. Given an integral polytope, say as its list of vertices, find all of its integral summands.

Again, this problem seems hard, but we shall give in Section 4 an algorithm for polytopes of dimension two which is "best possible" in the sense that the running time is linearly related to the number of decompositions.

2.3 Some preliminary results

We shall need more properties of the Minkowski sum. The next result from [4] describes how the faces decompose in a Minkowski sum of polytopes; for its proof, see Ewald [2, Theorem 1.5], Grünbaum [13, Theorem 1, p. 317], or Schneider [24, Theorem 1.7.5].

Lemma 5. *Let $P = Q + R$ where Q and R are polytopes in $\mathbb{R}^n$. Then*

(a) Each face of P is a Minkowski sum of unique faces of Q and R.
(b) Let P_1 be any face of P and $c_1, \ldots, c_k$ all of its vertices. Suppose that $c_i = a_i + b_i$ where $a_i \in Q$ and $b_i \in R$ for $1 \leq i \leq k$. Let

$$Q_1 = conv(a_1, \ldots, a_k), \quad R_1 = conv(b_1, \ldots, b_k).$$

Then Q_1 and R_1 are faces of Q and R, respectively, and $P_1 = Q_1 + R_1$.

A polytope of dimension two is called a *polygon*. (We refrain from using the term Newton polygon for a 2-dimensional Newton polytope as in number theory this term is used to refer to the lower boundary of the "Newton polyhedron" of certain power series.) The only proper faces of a polygon are its vertices and edges. For polygons, the above lemma can be rephrased as follows.

Corollary 6. *Let P, Q and R be convex polygons (in $\mathbb{R}^n$) with $P = Q + R$. Then every edge of P decomposes uniquely as the sum of an edge of Q and an edge of R, possibly one of them being a point. Conversely, any edge of Q or R is a summand of exactly one edge of P.*

3 Indecomposable polytopes

First of all, we mention the following two constructions of indecomposable polytopes from [4].

Theorem 7. *Let Q be any integral polytope in $\mathbb{R}^n$ contained in a hyperplane H and $v \in \mathbb{R}^n$ an integral point lying outside of H. Suppose that $v_1, \ldots, v_k$ are all the vertices of Q. Then the polytope $conv(v, Q)$ is integrally indecomposable iff*

$$\gcd(v - v_1, \ldots, v - v_k) = 1.$$

Here and hereafter the gcd of a collection of integral vectors is defined to be the gcd of all their coordinates together.

Theorem 8. *Let Q be an indecomposable integral polytope in $\mathbb{R}^n$ that is contained in a hyperplane H and has at least two points, and let $v \in \mathbb{R}^n$ be a point (not necessarily integral) lying outside of H. Let S be any set of integral points in the polytope $conv(v, Q)$. Then the polytope $conv(S, Q)$ is integrally indecomposable.*

The first construction shows that an integral line segment $\mathrm{conv}(v_0, v_1)$ is indecomposable iff $\gcd(v_0 - v_1) = 1$, and an integral triangle $\mathrm{conv}(v_0, v_1, v_2)$ is integrally indecomposable iff $\gcd(v_0 - v_1, v_0 - v_2) = 1$. The second construction gives many indecomposable polygons with more than three edges. These two constructions can be used iteratively to get indecomposable polytopes of any higher dimension.

In the following, we give a new construction based on a projection. Intuitively, one hopes that if a projection of a polytope is indecomposable then the polytope is indecomposable itself. Unfortunately, this is not true in general; consider for example a square and project it along one of its edges. The following lemma, however, gives a sufficient condition. We say that a linear map $\pi : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is *integral* if it maps integral points in $\mathbb{R}^n$ to integral points in $\mathbb{R}^m$. It is straightforward to see that the image of any integral polytope under an integral linear map is still an integral polytope.

Lemma 9. *Let P be any integral polytope in $\mathbb{R}^n$ and $\pi : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ any integral linear map. If $\pi(P)$ is integrally indecomposable and each vertex of $\pi(P)$ has only one preimage in P then P must be integrally indecomposable.*

Proof. It suffices to show that $\pi(P)$ is decomposable if P is decomposable. Suppose that $P = A + B$ for some integral polytopes A and B in $\mathbb{R}^n$ each with at least two points. Then $\pi(P) = \pi(A) + \pi(B)$. We need to show that both $\pi(A)$ and $\pi(B)$ have at least two points. Suppose otherwise, say $\pi(A)$ has only one point. Let w_0 be any vertex of P such that $\pi(w_0)$ is a vertex of $\pi(P)$. Since $P = A + B$, there are unique vertices $u_0 \in A$ and $v_0 \in B$ such that $w_0 = u_0 + v_0$. As A has at least two points, it has another vertex u_1 such that u_0u_1 is one of its edges. Then, by Lemma 5, P has an edge w_0w_1 that starts at w_0 and is parallel to u_0u_1 where w_1 is a vertex of P different from w_0. The latter property implies that $w_1 - w_0 = t(u_1 - u_0)$ for some real number t. Hence

$$\pi(w_1) - \pi(w_0) = \pi(w_1 - w_0) = \pi(t(u_1 - u_0)) = t(\pi(u_1) - \pi(u_0)) = 0,$$

as $\pi(A)$ has only one point and $u_1, u_0 \in A$. This means that π maps two vertices of P to one vertex of $\pi(P)$, contradicting our assumption.

Corollary 10. *Let P be any integral polytope in $\mathbb{R}^n$ and $\pi : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ any integral linear map that is injective on the vertices of P. If $\pi(P)$ is integrally indecomposable then so must be P.*

Theorem 11. *Let Q be any integrally indecomposable polytope in $\mathbb{R}^m$ and $\pi : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ any integral linear map. Let S be any set of integral points in $\pi^{-1}(Q)$ having exactly one point in $\pi^{-1}(v)$ for each vertex v of Q. Then the polytope conv(S) in $\mathbb{R}^n$ is integrally indecomposable.*

Proof. It follows directly from Lemma 9.

Remark. Theorem 8 can be viewed as a special case of Theorem 11 in the case that Q has sufficiently many integral points besides its vertices, since it seems likely that there is an integral linear map that projects integral points in the cone $\operatorname{conv}(v, Q)$ to integral points in its base Q. Such a projection is impossible if Q has no integral points other than its vertices.

In concluding this section, we would like to discuss the relationship of integral decomposibility with a different concept of decomposibility of polytopes defined in Grünbaum [13, Chapter 15]. Let P, Q be polytopes in $\mathbb{R}^n$ (not necessarily integral). We say that Q is *homothetic* to P if there is a real number $t \geq 0$ and a vector $a \in \mathbb{R}^n$ such that

$$Q = tP + a = \{tb + a : b \in P\}.$$

A polytope P is called *homothetically indecomposable* if it is the case that whenever $P = P_1 + P_2$ for any polytopes P_1 and P_2, then P_1 or P_2 is homothetic to P. Otherwise, P is called *homothetically decomposable.* Indecomposable polytopes in this sense have been extensively studied in the literature [3,14,19,20,25–27].

Homothetic decomposability is not directly comparable with integral decomposability. On the one hand, the only homothetically indecomposable polytopes in the plane are line segments and triangles so any polygon with more than 3 edges is homothetically decomposable [13,24]. On the other hand, we saw above that some triangles can be integrally decomposable and many polygons with more than 3 edges are integrally indecomposable! The next result, however, shows that homothetic indecomposability implies integral indecomposability under a simple condition.

Proposition 12. *Let Q be an integral polytope in $\mathbb{R}^n$ with vertices v_i, where $0 \leq i \leq k$. If Q is homothetically indecomposable and*

$$\gcd(v_0 - v_1, \cdots, v_0 - v_k) = 1,$$

then Q is integrally indecomposable.

Proof. Suppose that $Q = T + S$ for some integral polytopes T and S. Then T or S is homothetic to Q, say T. This means that there is a real number $r \geq 0$ and $a \in \mathbb{R}^n$ such that $T = rQ + a$. Hence the vertices of T are

$$u_i := rv_i + a, \quad i = 0, 1, \ldots, k.$$

Since T is integral, all the vertices $u_0, u_1, \ldots, u_k$ are integral and in particular

$$u_0 - u_i = r(v_0 - v_i), \quad i = 1, \ldots, k$$

are integral. So r must be a rational number and the denominator of r divides $\gcd(v_0 - v_1, \cdots, v_0 - v_k) = 1$; hence r is an integer. As $0 \leq r \leq 1$, we have $r = 0$ or 1. In either case, T is a trivial summand of Q. Therefore Q is integrally indecomposable.

By the above theorem, the homothetically indecomposable polytopes constructed in [3,14,19,20,26,27] give many integrally indecomposable polytopes.

4 Decomposing polytopes

In this section we present our algorithms for both testing polytopes for indecomposability and constructing summands of polytopes. We restrict our attention to polygons in Section 4.1 before considering the more general case in Section 4.2.

4.1 Polygons

Given a convex polygon in the Euclidean plane, one may form a finite sequence of vectors associated with it as follows. Let $v_0, v_1, \ldots, v_{m-1}$ be the vertices of the polygon ordered cyclically in a clockwise direction. The edges of P are represented by the vectors $E_i = v_i - v_{i-1} = (a_i, b_i)$ for $1 \leq i \leq m$, where $a_i, b_i \in \mathbb{Z}$ and the indices are taken modulo m. We call each E_i an *edge vector*. A vector $v = (a, b) \in \mathbb{Z}^2$ is called a *primitive vector* if $\gcd(a, b) = 1$. Let $n_i = \gcd(a_i, b_i)$ and define $e_i = (a_i/n_i, b_i/n_i)$. Then $E_i = n_i e_i$ where e_i is a primitive vector, $1 \leq i \leq m$. Each edge E_i contains precisely $n_i + 1$ integral points including its end points. The sequence of vectors $\{n_i e_i\}_{1 \leq i \leq m}$, which we call the *edge sequence* or a *polygonal sequence*, uniquely identifies the polygon up to translation determined by v_0, and will be the input to our polygon decomposition algorithm. It will be convenient to identify sequences with those obtained by extending the sequence by inserting an arbitrary number of zero vectors. We may thus assume that the edge sequence of a summand of a polygon P has the same length as that of P. As the boundary of the polygon is a closed path, we have that $\sum_{1 \leq i \leq m} n_i e_i = (0, 0)$.

Lemma 13. *Let P be a polygon with edge sequence $\{n_i e_i\}_{1 \leq i \leq m}$ where $e_i \in \mathbb{Z}^2$ are primitive vectors. Then an integral polygon is a summand of P iff its edge sequence is of the form $\{k_i e_i\}_{1 \leq i \leq m}$, $0 \leq k_i \leq n_i$, with $\sum_{1 \leq i \leq m} k_i e_i = (0, 0)$.*

Proof. Let $\{e'_i\}_{1 \leq i \leq m}$ be the edge sequence of an integral summand Q of P. By the final statement in Corollary 6, each edge of Q occurs as the summand of some edge ne of P where e is a primitive vector, and it is easily seen that its corresponding edge vector must be of the form ke with $0 \leq k \leq n$. The sum is zero simply because the boundary of Q is a closed path. Conversely, any sequence of this form will determine a closed path. Since $\{n_i e_i\}_{1 \leq i \leq m}$ is a polygonal sequence, $\{k_i e_i\}_{1 \leq i \leq m}$ must define the boundary of a convex polygon. It will be a summand of P, with the other summand having edge sequence $\{(n_i - k_i) e_i\}_{1 \leq i \leq m}$.

Given as input a sequence of edge vectors $\{n_i e_i\}_{1 \leq i \leq m}$ of a polygon P, our polygon decomposition algorithm will check for the existence of a sequence of integers k_i with $0 \leq k_i \leq n_i$, $1 \leq i \leq m$, such that $\sum_{1 \leq i \leq m} k_i e_i = (0, 0)$, $k_m \neq n_m$, and not all $k_i = 0$. (If P is decomposable then at least one of its summands has $k_m \neq n_m$.) Thus the decision problem underlying our algorithm is

Polygon Decomposability (PolyDecomp)
Input: The egde sequence $\{n_i e_i\}_{1\leq i\leq m}$ of an integral convex polygon P.
Question: Does P have a proper integral decomposition?

The input size of an instance of this problem is $O(m(\log N + \log E))$ where $N = \max\{n_1, \ldots, n_m\}$ and E the maximum of absolute values of the coordinates of e_i, $1 \leq i \leq m$. The next result puts the difficulty of this problem in context.

Proposition 14. PolyDecomp *is* NP*-complete.*

Proof. Certainly the language associated with PolyDecomp lies in NP as we may use a proper decomposition of P to verify membership of the language. We give a polynomial reduction of Partition to PolyDecomp which proves, since Partition is NP-complete [6], that PolyDecomp is NP-complete.

Recall that the input to Partition is a sequence $\{s_i\}_{1\leq i\leq m}$ of positive integers which we may take to be non-decreasing. Thus $s_1 \leq s_2 \leq \ldots \leq s_m$. Let $t = \sum_{1\leq i\leq m} s_i$. The question in Partition is whether there is a subsequence of $\{s_i\}$ with sum $t/2$. Observe that we may assume that t is even, for otherwise the question is easily answered. Consider now the following instance of PolyDecomp: the edge sequence

$$(s_1, 1), (s_2, 1), \ldots (s_m, 1), m(0, -1), (-t/2, -1), (-t/2, 1)$$

where all $n_i = 1$. Firstly, it is easy to check that this is indeed a polygonal sequence. Secondly, any polygon associated with the polygonal sequence has a proper decomposition if and only if the sequence $\{s_i\}_{1\leq i\leq m}$ has a subsequence with sum $t/2$. Thus we have a polynomial reduction, which completes the proof.

Since it is widely believed that NP $\neq$ P, it seems unreasonable to attempt to find a genuinely efficient algorithm for solving PolyDecomp; however, we shall present an algorithm below whose running time is polynomial in the length of the sides of the polygon rather than the logarithm of the lengths. In the parlance of [6], this is an example of a "pseudopolynomial-time" algorithm. In Section 5 we shall indicate how this algorithm may be used to test bivariable polynomials for absolute irreducibility; the algorithm thus obtained is efficient in terms of the total degree of the polynomial, rather than the number of non-zero terms. Thus the distinction between genuinely efficient algorithms for deciding polytope decomposability and "pseudopolynomial-time" algorithms is mirrored to a certain extent in that between efficient algorithms for polynomials in terms of their sparse and dense representations.

Algorithm 15 (PolyDecomp)
Input: The edge sequence $\{n_i e_i\}_{1 \le i \le m}$ of an integral convex polygon P starting at a vertex v_0 where $e_i \in \mathbb{Z}^2$ are primitive vectors.
Output: Whether P is decomposable.

Step 1: Compute the set IP of all the integral points in P, and set $A_0 = \emptyset$.

Step 2: For i from 1 up to $m-1$, compute the set A_i of points in IP that are reachable via the vectors $e_1, \ldots, e_i$:

2.1 For each $0 < k \le n_i$, if $v_0 + ke_i \in IP$ then add it to A_i;
2.2 For each $u \in A_{i-1}$ and $0 \le k \le n_i$, if $u + ke_i \in IP$ then add it to A_i.

Step 3: Compute the last set A_m: For each $u \in A_{m-1}$ and $0 \le k < n_m$, if $u + ke_m \in IP$ then add it to A_m.

Step 4: Return "Indecomposable" if $v_0 \notin A_m$ and "Decomposable" otherwise.

Theorem 16. *The above algorithm decides decomposability correctly in $O(tmN)$ vector operations where t is the number of integral points in P, m the number of edges and N the maximum number of integral points on an edge.*

Proof. The running time is easy to see as each set A_i has size at most t. (Note that by a vector operation we mean addding two vectors, multiplying a vector by a scalar, or adjoining a point to a set.) To prove the correctness, observe that all the points in A_m are of the form $v_0 + \sum_{i=1}^m k_i e_i$, $0 \le k_i \le n_i$. Step 2.1 ensures that $k_i \ne 0$ for some $i < m$ and Step 3 insists that $k_m < n_m$ (note that $v_0 + ke_m \notin IP$ for all $k > 0$). If one of the points in A_m is equal to v_0 then $\sum_{i=1}^m k_i e_i = (0,0)$, and so the sequence $\{k_i e_i\}$ forms the edge sequence of a proper integral summand of P. On the other hand, for any proper integral summand Q of P, Q can be "slid" into P at v_0, that is, Q can be translated so that v_0 is a vertex of Q and Q lies inside P. Hence all the vertices of Q must lie in P and thus in IP. Consequently its edge sequence will be detected by our algorithm.

We next give a simple generalisation of the above algorithm which not only outputs the number of proper decompositions of the polygon, but also outputs an array. The array may then be used to recover all decompositions, a single "recovery" requiring linear time. Thus the total time taken to recover all decompositions is essentially linearly related to the number of decompositions. This is the best that one can expect; however, it does not yield a "pseudopolynomial-time" algorithm as the number of decompositions may be exponential in the area of the polygon. For example, consider the polygon with edge sequence

$$(1,1), (2,1), \ldots, (m,1), m(0,-1), t(-1,0)$$

where $t = (m+1)m/2$. The polygon has area less than $1^2 + 2^2 + \cdots + m^2 = O(m^3)$ while the number of integral summands is exactly 2^m.

Algorithm 17 (PolyDecompNum)
Input: The edge sequence $\{n_i e_i\}_{1 \le i \le m}$ of an integral convex polygon P starting at a vertex v_0 where $e_i \in \mathbb{Z}^2$ are primitive vectors.
Output: The number of integral summands of P including the trivial ones, and an array A. Each cell in A contains a pair (u, S) where u is a non-negative integer and S is a subset of $\{(k, i) : 1 \le k \le n_i, 1 \le i \le m\}$.

Step 1: Compute the set IP of all the integral points in P (so $v_0 \in IP$); say IP has t points. Initialize a t-array A_0 indexed by the points in IP. Set $A_0[v] := (0, \emptyset)$ for all $v \in IP$ except the cell $A_0[v_0]$ which is set to $(1, \emptyset)$.

Step 2: For i from 1 up to m, compute the t-array A_i from A_{i-1}:

2.1 First copy the contents of all the cells of A_{i-1} into A_i (this step is for $k = 0$).
2.2 For each $v \in IP$ with the first number of the cell $A_{i-1}[v]$ nonzero, and for each $0 < k \le n_i$, if $v' = v + ke_i \in IP$ then update the cell $A_i[v']$ as follows: if (u_1, S_1) is the value of $A_{i-1}[v]$ and (u_2, S_2) the current value of $A_i[v']$ then the new value of $A_i[v']$ is $(u_1 + u_2, S_2 \cup \{(k, i)\})$.

Step 3: Return the number u and the array $A = A_m$, where (u, S) is the content of cell $A_m[v_0]$.

Theorem 18. *The integer output by Algorithm 17 is the total number of integral summands of the polygon P.*

Proof. Supposing $v = v_0 + k_1 e_1 + \cdots + k_i e_i$, we may view the vector sum as a path from v_0 to v, so the number of such paths is equal to the sum of the numbers of paths from v_0 to $v - ke_i$ for $0 \le k \le n_i$, using $e_1, \ldots, e_{i-1}$. Hence the numbers of paths can be computed iteratively as described in the algorithm: the number u in $A_i[v]$ records the number of paths from v_0 to v using $e_1, \ldots, e_i$ and the set S records all the pairs (k, j), $j \le i$, for which a path reaches v with its last edge being ke_j with $k > 0$. Thus the integer in cell $A_m[v_0]$ is the total number of closed paths $\sum_{1 \le i \le m} k_i e_i$ starting at v_0. By Lemma 13 this is the number of integral summands of P.

The significance of the array A output by the algorithm is that it may be used to recover all decompositions of the polygon P. We show how a single decomposition can be recovered: Suppose the cell $A[v_0]$ contains the pair (u, S). Choose any $(k, i) \in S$. The line segment ke_i will be the "final edge" (counting clockwise) in our summand of P. Let (u', S') be the contents of cell $B[v_0 - ke_i]$. Pick any $(k', i') \in S'$ with $i' < i$. The line segment $k'e_{i'}$ will be the "penultimate edge" in our summand of P. We continue in this way, and as our sequence of i's is decreasing we shall eventually return to the cell $A[v_0]$. At that point we will have recovered one summand in a decomposition of P.

With regard to the running time, each cell in the array can be updated at most mN times, thus the running time is $O(tmN)$ "cell updates". The data in each cell is a pair (u, S) where S is a set of size at most mN and u an integer less than N^m (an upper bound on the number of summands). Updating the integer u involves integer addition and this has a bit complexity of $O(\log N^m) = O(m \log N)$. Updating the set S simply involves unioning it with an element (k, i). Ignoring logarithmic factors, we can consider this a single bit operation. Thus the running time of **PolyDecompNum** is $O(tm^2N)$ bit operations, ignoring logarithmic factors.

4.2 Higher dimensional polytopes

The problem of testing higher dimensional polytopes for decomposability appears to be significantly more difficult. Certainly it is NP-complete as it includes that of polygons as a special case. It would be interesting to investigate whether this problem was "strongly NP-complete" in the sense of [6]; this essentially means that the problem remains "NP-complete" when one bounds running time by the lengths, instead of logarithm of the lengths, of the edge vectors. If this more general problem is "strongly NP-complete" then it is unlikely there is an algorithm for determining whether a convex polytope of arbitrary dimension is indecomposable whose running time is polynomial in terms of the volume of the polytope.

In this section, we present a heuristic "randomized algorithm" based on the projections considered in Lemma 9. The algorithm has running time polynomial in the lengths of the edges of the polytope, thus is "efficient" in the sense which we have been considering. The idea is to choose a random integral linear map that projects a polytope into a polygon in a plane and then test the decomposability of the polygon. If the polygon is indecomposable and the condition of Lemma 9 is satisfied then the original polytope is indecomposable. We will show that the condition of Lemma 9 is always satisfied with high probability, but we do not know how to prove a good bound on the probability that the projected polygon be indecomposable when the original polytope is indecomposable.

We now describe the details of our algorithm. Let $S \subset \mathbb{R}^n$ be any finite set of integral points, which will be the input to our algorithm, and $P = \mathrm{conv}(S)$. We want to decide whether P is integrally indecomposable. Note that P can be computed from S by any of the algorithms in [9,10]; however, our algorithm does not require that the vertices, which are all in S, of P be known in advance but detects them automatically. This is because the points of S that are mapped to vertices of a polygon will be vertices of P, provided each vertex of the polygon has only one preimage in S.

To describe a projection, we write points in $\mathbb{R}^n$ as column vectors, so a set S of ℓ points can be represented as an $n \times \ell$ matrix where each column stands for a point; for convenience, we still denote the matrix by S. As the points in S are distinct so are the columns of S. Let $u, v \in \mathbb{R}^n$ be two integral points. Then for any point $w \in \mathbb{R}^n$, the matrix-vector product $(u,v)^t w$ can be viewed as a point in $\mathbb{R}^2$. This defines an integral projection π from $\mathbb{R}^n$ into $\mathbb{R}^2$ and

$$(u,v)^t S \tag{1}$$

is the image of S under π in $\mathbb{R}^2$. The polygon defined by the convex hull of the points in (1) is called the *shadow polygon*, or simply *shadow*, of P projected by u and v. The next lemma from [5] arises in a different context and tells us how likely it is that the projection is injective on the set S; its proof is straightforward.

Lemma 19. *Let S be an $n \times \ell$ matrix over a field with no repeated columns and let K be any subset of cardinality k of the same field. Pick $u_i \in K$ randomly and independently, $1 \le i \le n$, and let*

$$(a_1, \cdots, a_\ell) = (u_1, \cdots, u_n)S.$$

Then with probability at least $1 - \frac{\ell(\ell-1)}{2k}$ the entries $a_1, \ldots, a_\ell$ are distinct.

Now let $K = \{-\ell^2, \ldots, -1, 0, 1, \ldots, \ell^2\}$ which has $k = 2\ell^2 + 1$ integers. If we choose the entries of u and v from K at random and independently, then with probability at least $3/4$ the points in (1) are distinct, so the condition in Lemma 9 is satisfied, i.e., each vertex of the shadow has only one preimage in S. This probability can be increased arbitrarily close to 1 if one increases the size of the set K.

Algorithm 20 (PolytopeDecomp)
Input: A finite set S of integral points in $\mathbb{R}^n$.

Output: "Indecomposable"or "Failure"; the first case means that the polytope $P = \mathrm{conv}(S)$ is proved to be indecomposable while the latter means the decomposibility of P is not decided.

Step 0: Form the points in S as an $n \times \ell$ matrix, still denoted by S, where ℓ is the cardinality of S and each column represents a point. Fix a set K of small integers.

Step 1: Pick two vectors $u, v \in K^n$ randomly and compute the projection $(u,v)^t S = (a_1, \ldots, a_\ell)$ where $a_i \in \mathbb{Z}^2$.

Step 2: Compute the vertices, say $v_1, \ldots, v_m$ in a clockwise direction, of the convex polygon defined by the points $a_1, \ldots, a_\ell$. If more than two points of S are mapped to one of the vertices v_i's, then output "Failure" and stop here.

Step 3: Compute $E_i = v_i - v_{i-1} = n_i e_i$ where n_i is a positive integer and e_i is a primitive vector, $1 \leq i \leq m$.

Step 4: Input the edge sequence $\{n_i e_i\}$ to Algorithm **PolyDecomp**. If the latter says "Indecomposable" then output "Indecomposable", otherwise output "Failure".

The correctness of this algorithm follows from our discussion above. If P is integrally decomposable then the algorithm will always output "Failure". It remains an open problem to determine how likely it is that the algorithm will output "Indecompsable" if P is integrally indecomposable. It is possible that there are indecomposable polytopes whose shadow polygons are always decomposable; for such polytopes our algorithm will not work. We would be very interested in seeing such examples.

On the other hand, it can be proved that most polytopes in $\mathbb{R}^n$, $n \geq 3$, are homothetically indecomposable [24, Theorem 3.2.14, p152]. By Proposition 12, we may expect that most integral polytopes are integrally indecomposable so our algorithm may detect most of them quickly. It would be interesting to know how likely it is that a random shadow polygon of a random integral polytope (under some probability distribution) is indecomposable.

5 Applications to polynomials

A direct application of Algorithm 15 in the light of Corollary 2 gives an algorithm for testing absolute irreducibility of bivariable polynomials. One simply first checks whether the input polynomial has any factors of the form X_i and if not computes the edge sequence of its Newton polytope, which can be done in $O(t \log t)$ operations where t is the number of nonzero terms in the polynomial. Algorithm 15 may then be used to determine whether this polygon is decomposable; if it is indecomposable then the polynomial must be absolutely irreducible. In the case that the polygon is decomposable the test is inconclusive. The running time of this algorithm is easily checked to be $O(n^3)$ where n is the total degree of the polynomial. A similar test based on Algorithm 20 may be devised to test general multivariable polynomials for absolute irreducibility where S is taken to be the set of support vectors of the polynomial to be tested.

Certainly, this polytope approach cannot decide irreducibility of some polynomials since it uses only their "shapes", i.e. Newton polytopes, and the coefficients do not come into play. However, our algorithm is extremely fast compared to the infallible algorithms in [1,7,11,15,17,18], thus it may be used as a pretest before applying the more expensive methods. For random sparse polynomials, their Newton polytopes may be viewed as random integral polytopes. As we mentioned at the end of the last section, most integral polytopes are expected to be indecomposable. Hence the "shapes" of most polynomials are indecomposable, so our algorithm can detect them quickly in most of the cases. This means that our polytope method should be particularly effective for random sparse polynomials.

We finish by returning to the problem of Sturmfels quoted in Section 1. In this problem, one is given the list of support vectors of a polynomial f but the coefficients of f are not specified. From the support vectors, one can compute their convex hull. So one is essentially given the Newton polytope P_f of f with the requirement that the terms of f corresponding to the integral points of P_f not on the given list of support vectors must have zero coefficient. The question is how such a polynomial f factors in general? What are the Newton polytopes and support vectors for the factors?

A natural approach is to find the set of all integral summands of P_f, as this set contains the Newton polytopes of all possible factors. Each summand may correspond to a factor of f, and if this is the case then the set of integral points in the summand contains the support vectors of the corresponding factor. For bivariable polynomials, one may find all integral summands by applying Algorithm 17 and the method suggested immediately after it. It seems that most integral polytopes do not have many integral summands, so our method is expected to be effective for random sparse polynomials. We would like to add that this method can be refined by taking into account the possible factorizations of the univariable polynomials defined by the edges of the polygon; however, we do not pursue this at present.

We should point out that some integral summands may not correspond to any factor of f. For example, let

$$f = (a + bX^n) + Y^m(c + dX^n) \in K[X, Y].$$

Its Newton polytope is a rectangle defined by the support vectors $(0, 0)$, $(0, n)$, $(m, 0)$ and (n, m). This rectangle has $(n+1)(m+1)$ integral summands. But f is almost always absolutely irreducible except for a few cases! Moreover, in general even when we find a summand P_g of P_f which corresponds to a factor g of the polynomial f under consideration, it may be the case that not all integral points in P_g are support vectors of g. We only know for sure that the vertices of P_g are among the support vectors of g.

Finally, we mention that deciding irreducibility of sparse polynomials can be considered a special case of the above problem. Even though we have shown that deciding decomposability of integral polytopes is NP-complete, we still do not know whether deciding irreducibility is also NP-complete. The latter problem is not even known for sparse univariable polynomials.

6 Conclusion

The Newton polytope of a polynomial carrys a lot of information about its factors, and so it is fruitful to study algorithms for deciding decomposability of integral polytopes and for finding all the integral summands when they are decomposable. For polygons, we showed that deciding decomposability is

NP-complete but gave a pseudo-polynomial time algorithm for testing decomposability and for constructing all possible decompositions. For polytopes of dimension larger than two, we presented an indecomposability lemma based on projections, and this lemma gives a heuristic method for testing decomposability of polytopes in any dimension. However, a rigorous analysis of this algorithm is still lacking. It is also desirable to have an algorithm for finding all the integral summands for polytopes in arbitrary dimensions. The corresponding problems for (sparse) polynomials are also open: it is not even known whether deciding irreducibility of sparse polynomials is NP-complete.

References

1. D. Duval, Absolute factorization of polynomials: a geometric approach, *SIAM J. Comput.* **20** (1991), 1–21.
2. G. Ewald, Combinatorial Convexity and Algebraic Geometry, GTM 168, Springer 1996.
3. D. Gale, Irreducible convex sets, *Proc. Intern. Congr. Math.*, Amsterdam 1954, Vol. 2, 217–218.
4. S. Gao, Absolute irreducibility of polynomials via Newton polytopes, preprint, 1998.
5. S. Gao, Factoring multivariate polynomials via PDE, in preparation.
6. M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, New York, 1979.
7. J. von zur Gathen and E. Kaltofen, Factorization of multivariate polynomials over finite fields, *Math. Comp.* **45** (1985), no. 171, 251–261.
8. I. Gel'fand, M. Kapranov and A. Zelevinsky, Discriminants, Resultants and Multi-Dimensional Determinants, Birkhäuser, Boston, 1994.
9. J.E. Goodman and J. O'Rourke (Eds), Handbook of Discrete and Computational Geometry, Elsevier Science, Amsterdam, 1997.
10. R. L. Graham, An efficient algorithm for determining the convex hull of a finite planar set, *Inform. Process. Lett.* **1** (1972), 132–133.
11. D. Yu Grigoryev and A. L. Chistov, Fast factorization of polynomials into irreducible ones and the solution of systems of algebraic equations, *Dokl. Akad. Nauk SSSR* **275** (1984), no. 6, 1302–1306. English translation: *Soviet Math. Dokl.* **29** (1984), no. 2, 380–383.
12. P. M. Gruber and J. M. Wills (Eds), Handbook of Convex Geometry, Vol. A and B, Elsevier Science, Amsterdam, 1993.
13. B. Grünbaum, Convex Polytopes, Interscience Publ., London, New York, Sydney, 1967.
14. M. Kallay, Indecomposable polytopes, *Israel J. Math.* **41** (1982), no. 3, 235–243.
15. E. Kaltofen, Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization, *SIAM J. Comput.* **14** (1985), no. 2, 469–489.
16. E. Kaltofen, Polynomial factorization 1987–1991, in *LATIN '92* (São Paulo, 1992), 294–313, Lecture Notes in Comput. Sci., 583, Springer, Berlin, 1992.
17. A. K. Lenstra, Factoring multivariate polynomials over finite fields, *J. Comput. System Sci.* **30** (1985), no. 2, 235–248.
18. A. K. Lenstra, Factoring multivariate polynomials over algebraic number fields, *SIAM J. Comput.* **16** (1987), no. 3, 591–598.

19. P. McMullen, Indecomposable convex polytopes, *Israel J. Math.* **58** (1987), no. 3, 321–323.
20. W. Meyer, Indecomposable polytopes, *Trans. Amer. Math. Soc.* **190** (1974), 77–86.
21. A. M. Ostrowski, Über die Bedeutung der Theorie der konvexen Polyeder für die formale Algebra, *Jahresberichte Deutsche Math. Verein* **30** (1921), 98–99.
22. A. M. Ostrowski, On multiplication and factorization of polynomials, I. Lexicographic ordering and extreme aggregates of terms, *Aequationes Math.* **13** (1975), 201–228.
23. A. M. Ostrowski, On multiplication and factorization of polynomials, II. Irreducibility discussion, *Aequationes Math.* **14** (1976), 1–32.
24. R. Schneider, Convex bodies: the Brunn-Minkowski theory, Encyclopedia of Mathematics and its Applications, 44. Cambridge University Press, Cambridge, 1993.
25. G. C. Shephard, Decomposable convex polyhedra, *Mathematika* **10** (1963), 89–95.
26. Z. Smilansky, An indecomposable polytope all of whose facets are decomposable, *Mathematika* **33** (1986), no. 2, 192–196.
27. Z. Smilansky, Decomposability of polytopes and polyhedra, *Geometriae Dedicata* **24** (1987), no. 1, 29–49.
28. B. Sturmfels, Gröbner Bases and Convex Polytopes, University Lecture Series, Vol. 8. American Mathematical Society, 1996.
29. B. L. van der Waerden, Modern Algebra, Vol. 1, Ungar, New York, 1953.