# Efficient $q$-Integer Linear Decomposition of Multivariate Polynomials

Mark Giesbrecht

*Symbolic Computation Group, Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, N2L 3G1, Canada*

Hui Huang

*School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning, 116024, China*

George Labahn

*Symbolic Computation Group, Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, N2L 3G1, Canada*

Eugene Zima

*Physics and Computer Science, Wilfrid Laurier University, Waterloo, ON, N2L 3C5, Canada*

## Abstract

We present two new algorithms for the computation of the $q$-integer linear decomposition of a multivariate polynomial. Such a decomposition is essential in the $q$-analogous world of symbolic summation, for example, describing the $q$-counterpart of Ore-Sato theory or determining the applicability of the $q$-analogue of Zeilberger's algorithm to a $q$-hypergeometric term. Both of our algorithms require only basic integer and polynomial arithmetic and work for any unique factorization domain containing the ring of integers. Complete complexity analyses are conducted for both our algorithms and two previous algorithms in the case of multivariate integer polynomials, showing that our algorithms have better theoretical performances. A Maple implementation is also included which suggests that our algorithms are also much faster in practice than previous algorithms.

*Keywords:* $q$-Analogue, Integer-linear polynomials, Polynomial decomposition,
Newton polytope, Creative telescoping, Ore-Sato theory

## 1. Introduction

Many objects in the ordinary shift world of symbolic summation find a natural counterpart commonly called *q-analogues*. In a typical situation, these are just slight adaptations of the

*Email addresses:* `mwg@uwaterloo.ca` (Mark Giesbrecht), `huanghui@dlut.edu.cn` (Hui Huang), `glabahn@uwaterloo.ca` (George Labahn), `ezima@wlu.ca` (Eugene Zima)

original objects but with involved variables promoted to exponents of an additional parameter $q$. Techniques for handling the originals often carry over to their $q$-analogues with some subtle modifications.

In this paper, we deal with the $q$-analogue of integer-linear decompositions of polynomials and aim to provide an intensive treatment for its computation in analogy to (Giesbrecht et al., 2019). Surprisingly, although this $q$-analogue is obtained by modeling its ordinary shift counterpart, the primary technique used in (Giesbrecht et al., 2019) can not be easily adapted to compute it due to different structures. A new alternative technique will be presented in this $q$-shift case.

In order to describe more details, we let $\mathsf{D}$ be a ring of characteristic zero and let $\mathsf{R} = \mathsf{D}[q, q^{-1}]$ be its transcendental ring extension by the indeterminate $q$. For $n$ discrete indeterminates $k_1, \ldots, k_n$ distinct from $q$, we know that $q^{k_1}, \ldots, q^{k_n}$ are transcendental over $\mathsf{R}$. We can then consider polynomials in $q^{k_1}, \ldots, q^{k_n}$ over $\mathsf{R}$, all of which form a well-defined ring denoted by $\mathsf{R}[q^{k_1}, \ldots, q^{k_n}]$. We say an irreducible polynomial $p \in \mathsf{R}[q^{k_1}, \ldots, q^{k_n}]$ is *$q$-integer linear* over $\mathsf{R}$ if there exists a univariate polynomial $P \in \mathsf{R}[y]$ and two integer-linear polynomials $\sum_{i=1}^n \alpha_i k_i, \sum_{i=1}^n \lambda_i k_i \in \mathbb{Z}[k_1, \ldots, k_n]$ such that

$$p(q^{k_1}, \ldots, q^{k_n}) = q^{\sum_{i=1}^n \alpha_i k_i} P(q^{\sum_{i=1}^n \lambda_i k_i}).$$

In order to avoid superscripts, we will write the indeterminates $q^{k_1}, \ldots, q^{k_n}$ as the variables $x_1, \ldots, x_n$ in the sequel of the paper. Then the above definition can be rephrased as follows. An irreducible polynomial $p \in \mathsf{R}[x_1, \ldots, x_n]$ is called *$q$-integer linear* over $\mathsf{R}$ if there exists a univariate polynomial $P \in \mathsf{R}[y]$ and integers $\alpha_1, \ldots, \alpha_n, \lambda_1, \ldots, \lambda_n$ such that

$$p(x_1, \ldots, x_n) = x_1^{\alpha_1} \cdots x_n^{\alpha_n} P(x_1^{\lambda_1} \cdots x_n^{\lambda_n}). \tag{1.1}$$

Note that the indeterminate $q$ is hidden in the variables $x_1, \ldots, x_n$. Since a common factor of the $\lambda_i$ can be pulled out and absorbed into $P$, and a monomial can be merged into $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ if necessary, we assume that the integers $\lambda_1, \ldots, \lambda_n$ have no common divisor, that the last nonzero integer is nonnegative, that $\lambda_i = 0$ whenever $\deg_{x_i}(p) = 0$ and that $P(0) \neq 0$. Such a vector $(\lambda_1, \ldots, \lambda_n)$, as well as such a polynomial $P$, is unique. We call the vector $(\lambda_1, \ldots, \lambda_n)$ the *$q$-integer linear type* of $p$ and the polynomial $P$ its *corresponding univariate polynomial*. Note that the resulting $\alpha_1, \ldots, \alpha_n$ all belong to $\mathbb{N}$ since $p \in \mathsf{R}[x_1, \ldots, x_n]$ and $P \in \mathsf{R}[y]$. A polynomial in $\mathsf{R}[x_1, \ldots, x_n]$ is called *$q$-integer linear* (over $\mathsf{R}$) if all its irreducible factors are $q$-integer linear, possibly with different $q$-integer linear types. For a polynomial $p \in \mathsf{R}[x_1, \ldots, x_n]$, we can define its *$q$-integer linear decomposition* by factoring into irreducible $q$-integer linear or non-$q$-integer linear polynomials and collecting irreducible factors having common types.

As with its ordinary shift counterpart, $q$-integer linear polynomials find broad applications in the $q$-analysis of symbolic summation. In particular, it is an important ingredient of the $q$-analogue of the Ore-Sato theorem for describing the structure of multivariate $q$-hypergeometric terms (Du and Li, 2019), which in turn, as indicated by (Chen and Koutschan, 2019), serves as a promising indispensable tool for settling a $q$-analogue of Wilf-Zeilberger's conjecture (Wilf and Zeilberger, 1992). Moreover, the $q$-integer linearity of polynomials plays a crucial role in detecting the applicability of the $q$-analogue of Zeilberger's algorithm (also known as the method of creative telescoping) for bivariate $q$-hypergeometric terms (Chen et al., 2005).

The full $q$-integer linear decomposition of polynomials is also very useful. On the one hand, it provides a natural way to determine the $q$-integer linearity of a given polynomial. On the other hand, it enables one to compute the $q$-analogue of Ore-Sato decomposition of a given $q$-hypergeometic term, and can also be employed to develop a fast creative telescoping algorithm for rational functions in the $q$-shift setting in analogy to (Giesbrecht et al., 2021). Evidently, the efficiency of the computation of $q$-integer linear decompositions directly affects the utility of all these algorithms.

In contrast to the ordinary shift case (Abramov and Le, 2002; Giesbrecht et al., 2019; Li and Zhang, 2013), algorithms for computing the $q$-integer linear decomposition of a multivariate polynomial are not very well developed. As far as we are aware, there is only one algorithm available to compute such a decomposition of a bivariate polynomial. This algorithm was first developed by Le (2001, §5) with an extended description provided in (Le et al., 2001). Except for using the same pattern as its ordinary shift counterpart (Abramov and Le, 2002), this algorithm takes use of a completely different strategy, especially for finding $q$-integer linear types. This is mainly because all $q$-integer linear types appear as the exponent vectors of $p$, rather than as the coefficients in the ordinary shift case. The main idea used by Le (2001, §5) is to first find candidates for $q$-integer linear types by computing a resultant and then, for each candidate, extract the corresponding univariate polynomial via bivariate GCD computations. Given the algebraic machinery on which the algorithm is based, it is not clear how one can directly generalize this to handle polynomials in more than two variables.

The main contribution of this paper is a pair of new fast algorithms for computing the $q$-integer linear decomposition of a multivariate polynomial. Both algorithms will work for any unique factorization domain containing all integers and for any polynomial with an arbitrary number of variables. The first approach combines the main ideas of the two previous algorithms - in the sense that it follows the pattern of the algorithm of Le and also makes use of exponent vectors as the factorization-based algorithm - but avoids the computation of resultants as well as the need for full irreducible factorization. More precisely, this approach reduces the problem of finding candidates for $q$-integer linear types to the well-studied geometric task of constructing the Newton polytope of the given polynomial, implying computations only using basic arithmetic operations $(+, -, \div, \times)$ of integers. It then computes each corresponding univariate polynomial by a content computation. As such we show that the $q$-analogue is actually simpler than its ordinary shift counterpart in the sense that, instead of finding rational roots of polynomials, one merely needs to perform basic integer manipulations.

Our second approach uses a bivariate-based approach. This scheme takes the bivariate version of our previous algorithm, that is, the algorithm for computing the $q$-integer linear decomposition of a bivariate polynomial, as a base case and iteratively tackles only two variables at a time until all variables are treated. Clearly, our two approaches coincide in the bivariate case.

Both approaches appear to be efficient in practice, though the second method shows an advantage for polynomials of a large number of variables. In order to do a theoretical comparison we have analyzed the worst-case running time complexity of both approaches in the case of multivariate polynomials over $\mathbb{Z}[q, q^{-1}]$. The analysis shows that the second approach is superior to the first one when the given polynomial has more than two variables. When restricted to the case of bivariate polynomials over $\mathbb{Z}[q, q^{-1}]$, the two approaches merge into one, which in turn is considerably faster than the algorithm of Le and the algorithm based on factorization. In addition, we also give experimental results which verify our complexity comparisons.

An additional contribution is to use our bivariate-based scheme (approach two) to extend the algorithm of Le so that it can readily tackle polynomials in any number of variables. For the sake of completeness, we also include another algorithm based on full irreducible factorization, which can be viewed as a $q$-analogue of the algorithm developed by Li and Zhang (2013). This algorithm makes use of the observation that the difference of exponent vectors of any two monomials appearing in an irreducible $q$-integer linear polynomial, say the polynomial $p$ of the form (1.1), must be a scalar multiple of the $q$-integer linear type $(\lambda_1, \ldots, \lambda_n)$. We also give a complexity analysis for both algorithms, at least in the case of bivariate polynomials over $\mathbb{Z}[q, q^{-1}]$. This supports the superiority of the factorization-based algorithm over the algorithm of Le. The same phenomenon was observed in the empirical tests.

The remainder of the paper proceeds as follows. Background and basic notions required in the paper are provided in the next section. Our two new approaches for computing $q$-integer linear decompositions of multivariate polynomials are given successively in Sections 3 and 4. The following section provides a complexity comparison of our two algorithms, the algorithm of Le and the factorization-based algorithm. The paper ends with an experimental comparison among all algorithms, along with a conclusion section.

## 2. Preliminaries: Polynomials and Newton polytopes

Throughout the paper, we let $\mathsf{D}$ be a unique factorization domain (UFD) of characteristic zero with $\mathsf{R} = \mathsf{D}[q, q^{-1}]$ denoting the transcendental ring extension by an indeterminate $q$. Note that a domain of characteristic zero always contains the ring of integers $\mathbb{Z}$ as a subdomain. Let $\mathsf{R}[x_1, \ldots, x_n]$ be the ring of polynomials in $x_1, \ldots, x_n$ over $\mathsf{R}$, where $x_1, \ldots, x_n$ are variables distinct from $q$. We reserve the variables $x$ and $y$ as synonyms for $x_1$ and $x_2$, respectively, so as to avoid subscripts in the case when $n \leq 2$.

Let $p$ be a polynomial in $\mathsf{R}[x_1, \ldots, x_n]$. Throughout this paper we will order monomials in $\mathsf{R}[x_1, \ldots, x_n]$ using a pure lexicographic order in $x_1 \prec \cdots \prec x_n$. For this order we let $\operatorname{lc}(p)$ and $\deg(p)$ denote the leading coefficient and the total degree, respectively, of $p$ with respect to $x_1, \ldots, x_n$. We follow the convention that $\deg(0) = -\infty$. We say that $p$ is *monic* (over $\mathsf{R}$) if $\operatorname{lc}(p) = 1$. The *content* of $p$ (over $\mathsf{R}$), denoted by $\operatorname{cont}(p)$, is the greatest common divisor (GCD) over $\mathsf{R}$ of the coefficients of $p$ with respect to $x_1, \ldots, x_n$ with $p$ being *primitive* if $\operatorname{cont}(p) = 1$. The *primitive part* $\operatorname{prim}(p)$ of $p$ (over $\mathsf{R}$) is defined as $p/\operatorname{cont}(p)$. For brevity, we will omit the domain if it is clear from the context. In certain instances, we also need to consider the above notions with respect to a subset of the $n$ variables. In these cases, we will either specify the relevant domain or indicate the related variables as subscripts of the corresponding notion. For example, $\operatorname{lc}_{x_1,x_2}(p)$, $\deg_{x_1,x_2}(p)$, $\operatorname{cont}_{x_1,x_2}(p)$ and $\operatorname{prim}_{x_1,x_2}(p)$ denote each function but applied to a polynomial $p$ viewing it as a polynomial in $x_1, x_2$ over the domain $\mathsf{R}[x_3, \ldots, x_n]$.

In order to obtain a canonical representation, we introduce the notion of $q$-primitive polynomials in the univariate case. A polynomial $p \in \mathsf{R}[y]$ is called *$q$-primitive* if it is primitive over $\mathsf{R}$ and its constant term $p(0)$ is nonzero. Note that this concept is a ring counterpart of $q$-monic polynomials introduced by Paule and Riese (1997). Clearly, any factor of a $q$-primitive polynomial in $\mathsf{R}[y]$ is again $q$-primitive.

The Newton polytope of multivariate polynomials plays a crucial role in our algorithms. In what follows, we recall some terminology and results on convex polytopes from a polynomial point of view. For a more general theory, one is referred to, for example, (Grünbaum, 2003).

In order to simplify notations, we employ bold letters, say $\boldsymbol{i}$, for a column vector $(i_1, \ldots, i_n)^T$ in the Euclidean space $\mathbb{R}^n$, and the *multi-index convention* $\boldsymbol{x}^{\boldsymbol{i}}$ for the monomial $x_1^{i_1} \cdots x_n^{i_n}$ if $\boldsymbol{i} \in \mathbb{Z}^n$. The zero vector in $\mathbb{R}^n$ is denoted by boldface $\mathbf{0}$. Taking advantage of this boldface notation, we later write $\mathsf{R}[\boldsymbol{x}]$ and $\mathsf{R}[\boldsymbol{x}, \boldsymbol{x}^{-1}]$ for the polynomial ring $\mathsf{R}[x_1, \ldots, x_n]$ and the Laurent polynomial ring $\mathsf{R}[x_1, x_1^{-1}, \ldots, x_n, x_n^{-1}]$, respectively.

Let $p \in \mathsf{R}[\boldsymbol{x}]$ be a polynomial of the form $\sum_{\boldsymbol{i}} a_{\boldsymbol{i}} \boldsymbol{x}^{\boldsymbol{i}}$ with $a_{\boldsymbol{i}} \in \mathsf{R}$, having finitely many nonzero terms. The *support* of $p$, denoted by $\operatorname{supp}(p)$, is defined as the set of indices $\boldsymbol{i} \in \mathbb{N}^n$ with the property that the corresponding coefficient $a_{\boldsymbol{i}}$ is nonzero. Clearly, $\mathrm{supp}(p)$ is a finite set in $\mathbb{N}^n$, and it is empty if and only if $p = 0$. An exponent vector $\boldsymbol{i}$ of $p$ can be considered as a point in $\mathbb{R}^n$. The convex hull of the set $\mathrm{supp}(p)$ in $\mathbb{R}^n$ is then known as the *Newton polytope* of $p$, denoted by $\mathrm{Newt}(p)$. By convention, $\mathrm{Newt}(0)$ is the empty set.

For two sets $A$ and $B$ in $\mathbb{R}^n$, their *Minkowski sum* is defined as the set

$$A + B = \{\boldsymbol{a} + \boldsymbol{b} \mid \boldsymbol{a} \in A, \boldsymbol{b} \in B\}.$$

The following well-known result, due to Ostrowski (1921, 1975), reveals the relation between the Newton polytope of a polynomial and those of its factors.

**Lemma 2.1** ((Ostrowski, 1921, 1975))**.** *Let $f, g \in \mathsf{R}[\boldsymbol{x}]$. Then* $\mathrm{Newt}(fg) = \mathrm{Newt}(f) + \mathrm{Newt}(g)$.

It proves convenient to extend the notion of Newton polytopes to Laurent polynomials in the ring $\mathsf{R}[\boldsymbol{x}, \boldsymbol{x}^{-1}]$. Notice that any Laurent polynomial from $\mathsf{R}[\boldsymbol{x}, \boldsymbol{x}^{-1}]$ can be written as the form $\boldsymbol{x}^{\boldsymbol{\alpha}} p$ for some $\boldsymbol{\alpha} \in \mathbb{Z}^n$ and $p \in \mathsf{R}[\boldsymbol{x}]$. Thus the *Newton polytope* of the given Laurent polynomial is defined to be the translation $\mathrm{Newt}(p) + \boldsymbol{\alpha}$ of $\mathrm{Newt}(p)$ by $\boldsymbol{\alpha}$. Evidently, Lemma 2.1 literally carries over to Laurent polynomials.

**Lemma 2.2.** *Let $f, g \in \mathsf{R}[\boldsymbol{x}, \boldsymbol{x}^{-1}]$. Then* $\mathrm{Newt}(fg) = \mathrm{Newt}(f) + \mathrm{Newt}(g)$.

We will consider faces of Newton polytopes. Let $C$ be a Newton polytope of a certain Laurent polynomial over $\mathsf{R}$. A hyperplane $H = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}^T \boldsymbol{x} = b\}$ with $\boldsymbol{a} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ and $b \in \mathbb{R}$ is called a *supporting hyperplane* of $C$ with *outward normal* $\boldsymbol{a}$ if $H \cap C \neq \emptyset$ and $\boldsymbol{a}^T \boldsymbol{x} \leq b$ for all $\boldsymbol{x} \in C$. We call the intersection $H \cap C$ a *face* of $C$. By convention, $\emptyset$ and $C$ are called *improper faces* of $C$. The faces of dimension zero and one are also called *vertices* and *edges*, respectively. Note that for any nonzero vector $\boldsymbol{a} \in \mathbb{R}^n$, there exists a unique supporting hyperplane of $C$ with outward normal $\boldsymbol{a}$ (cf. (Grünbaum, 2003, Theorem 8, Page 15)). We then refer to the intersection of this supporting hyperplane and $C$ as the face of $C$ determined by the outward normal $\boldsymbol{a}$.

**Lemma 2.3.** *Let $f, g \in R[\boldsymbol{x}, \boldsymbol{x}^{-1}]$ and $\boldsymbol{a} \in \mathbb{R}^n \setminus \{0\}$. Then $F_{fg,\boldsymbol{a}} = F_{f,\boldsymbol{a}} + F_{g,\boldsymbol{a}}$, where $F_{f,\boldsymbol{a}}$ is the face of* $\mathrm{Newt}(f)$ *determined by the outward normal $\boldsymbol{a}$.*

*Proof.* By Lemma 2.2, $\mathrm{Newt}(f) = \mathrm{Newt}(f) + \mathrm{Newt}(g)$. The assertion is then a direct result of (Grünbaum, 2003, Theorem 1, Page 317). □

## 3. $q$-Integer linear decomposition: the first approach

We are interested in finding the following decomposition of a polynomial, something briefly alluded to in the introduction.

**Definition 3.1.** *Let $p \in \mathsf{R}[\boldsymbol{x}]$ be a polynomial admitting the decomposition*

$$p = c\, \boldsymbol{x}^{\boldsymbol{\alpha}} P_0 \prod_{i=1}^{m} P_i(\boldsymbol{x}^{\boldsymbol{\lambda}_i}), \tag{3.1}$$

*where $c \in \mathsf{R}$, $m \in \mathbb{N}$, $\boldsymbol{\alpha} \in \mathbb{N}^n$, $\boldsymbol{\lambda}_i \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}$, $P_0 \in \mathsf{R}[\boldsymbol{x}]$ and $P_i \in \mathsf{R}[y]$. Then (3.1) is called the* $q$-integer linear decomposition *of $p$ (over $\mathsf{R}$) if*

*(1) $P_0$ is primitive and none of its irreducible factors of positive total degree is $q$-integer linear;*

*(2) each $P_i$ is q-primitive and of positive degree;*

*(3) each $\boldsymbol{\lambda}_i$ satisfies the conditions that $\gcd(\lambda_{i1}, \ldots, \lambda_{in}) = 1$ and its rightmost nonzero coordinate is positive* [1]*;*

*(4) the $\boldsymbol{\lambda}_i$ are pairwise distinct.*

*We call each $\boldsymbol{\lambda}_i$ a* $q$-integer linear type *of $p$ and $P_i$ its* corresponding univariate polynomial.

Evidently, $p$ is $q$-integer linear if and only if $P_0$ is a unit of $\mathsf{R}$ in (3.1). By full factorization, we see that every polynomial admits a $q$-integer linear decomposition. Moreover, this decomposition is unique up to the order of factors and multiplication by units of $\mathsf{R}$, according to the uniqueness of full factorization and that of the $q$-integer linear type of an irreducible polynomial.

Let $p \in \mathsf{R}[\boldsymbol{x}]$ be a polynomial of positive total degree. Without loss of generality, we assume that $p$ is primitive with respect to any variable from $\{x_1, \ldots, x_n\}$. Otherwise, we may replace $p$ by the remaining part after iteratively removing from $p$ its content with respect to $x_i$ for all $i = 1, \ldots, n$. Note that all these removed contents are polynomials over $\mathsf{R}$ having at most $(n-1)$ variables and hence can be dealt with recursively, knowing that univariate polynomials are all $q$-integer linear. With this set-up, $p$ admits the $q$-integer linear decomposition of the form (3.1), in which $c = 1$, $\alpha_n = 0$ and none of the types $\boldsymbol{\lambda}_i$ has zero coordinates. In order to compute such a decomposition, we mimic the strategy of Abramov and Le (2002) in the ordinary shift case, that is, we first find all possible candidates for $q$-integer linear types and then extract the corresponding univariate polynomial for each type.

### *3.1. Candidates for q-integer linear types*

Observe that all $q$-integer linear types $\boldsymbol{\lambda}_i$ in (3.1) appear as exponent vectors, and the Newton polytope of each $P_i(\boldsymbol{x}^{\boldsymbol{\lambda}_i})$ is just a line segment. This leads us to investigate edges of the Newton polytope of the given polynomial.

For this purpose, we assign a direction to each line segment in $\mathbb{R}^n$. Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ with $\boldsymbol{u} \neq \boldsymbol{v}$ and let $[\boldsymbol{u}, \boldsymbol{v}] = \{t\boldsymbol{u} + (1-t)\boldsymbol{v} \mid t \in \mathbb{R}, 0 \le t \le 1\}$ denotie the line segment connecting $\boldsymbol{u}, \boldsymbol{v}$. A nonzero vector $\boldsymbol{\lambda} \in \mathbb{R}^n$ is called the *direction vector* of $[\boldsymbol{u}, \boldsymbol{v}]$ if $\boldsymbol{u} - \boldsymbol{v} = t\boldsymbol{\lambda}$ for some $t \in \mathbb{R}$, $\gcd(\lambda_1, \ldots, \lambda_n) = 1$ and the rightmost nonzero coordinate of $\boldsymbol{\lambda}$ is positive. As before, the requirement of the positivity of the last nonzero coordinate guarantees the uniqueness of such a direction vector. Clearly, two parallel (nondegenerate) line segments share the same direction vector, and vice versa.

**Lemma 3.2.** *Let $p \in \mathsf{R}[\boldsymbol{x}] \setminus \mathsf{R}$ with $\mathrm{cont}_{x_1}(p) = \cdots = \mathrm{cont}_{x_n}(p) = 1$, and assume that it admits the q-integer linear decomposition (3.1). Then for any $i \in \mathbb{N}$ with $1 \le i \le m$, the Newton polytope of p possesses an edge of the direction vector $\boldsymbol{\lambda}_i$. Moreover, if $\mathrm{Newt}(p)$ is not a line segment then there are at least two such edges.*

*Proof.* There is nothing to show when $m = 0$, so assume that $m > 0$. We merely show the assertions for $i = m$, and then the lemma follows by symmetry.

Let $p^* = \boldsymbol{x}^{\boldsymbol{\alpha}} P_0 \prod_{i=1}^{m-1} P_i(\boldsymbol{x}^{\boldsymbol{\lambda}_i})$. Then $p^* \in \mathsf{R}[\boldsymbol{x}] \setminus \{0\}$, and by (3.1),

$$p = p^* P_m(\boldsymbol{x}^{\boldsymbol{\lambda}_m}). \tag{3.2}$$

[1] As mentioned in the introduction, the positivity of the rightmost nonzero coordinate of $\boldsymbol{\lambda}_i$ required here can be easily obtained and is used to make such a vector unique.

Notice that $\mathrm{Newt}(P_m(\boldsymbol{x}^{\boldsymbol{\lambda}_m}))$ is a line segment in $\mathbb{R}^n$ with direction vector $\boldsymbol{\lambda}_m$. Then for any nonzero vector $\boldsymbol{a} \in \mathbb{R}^n$ with $\boldsymbol{a}^T\boldsymbol{\lambda}_m = 0$, the supporting hyperplane of $\mathrm{Newt}(P_m(\boldsymbol{x}^{\boldsymbol{\lambda}_m}))$ determined by the outward normal $\boldsymbol{a}$ contains the whole polytope. This means that $\mathrm{Newt}(P_m(\boldsymbol{x}^{\boldsymbol{\lambda}_m}))$ itself is the (improper) edge determined by such an outward normal.

In order to show the first assertion, it then amounts to finding a nonzero vector $\boldsymbol{a} \in \mathbb{R}^n$ with $\boldsymbol{a}^T\boldsymbol{\lambda}_m = 0$ such that the face of $\mathrm{Newt}(p^*)$ determined by the outward normal $\boldsymbol{a}$ is either a vertex or an edge parallel to $\mathrm{Newt}(P_m(\boldsymbol{x}^{\boldsymbol{\lambda}_m}))$. The rest then follows by (3.2), Lemma 2.3 and the observation that the Minkowski sum of a line with a point or another parallel line is again a line parallel to the original line.

By an affine coordinate transformation if necessary, we may assume without loss of generality that $\boldsymbol{\lambda}_m$ is equal to the $n$-th unit vector $\boldsymbol{e}_n = (0, \ldots, 0, 1)^T \in \mathbb{R}^n$. Then $\mathrm{Newt}(P_m(\boldsymbol{x}^{\boldsymbol{\lambda}_m}))$ is contained by the $x_n$-axis. We now consider the projection of $\mathrm{Newt}(p^*)$ onto the hyperplane $\{\boldsymbol{x} \in \mathbb{R}^n \mid x_n = 0\}$ in the direction of $\boldsymbol{\lambda}_m = \boldsymbol{e}_n$, that is,

$$\mathrm{Proj}_n(p^*) = \{\boldsymbol{x} \in \mathbb{R}^n \mid x_n = 0 \text{ and } \boldsymbol{x} + t\boldsymbol{e}_n \in \mathrm{Newt}(p^*) \text{ for some } t \in \mathbb{R}\}.$$

This is again a Newton polytope by (Grünbaum, 2003, Theorem 8, Page 74). Since $p^*$ is nonzero, $\mathrm{Newt}(p^*)$ is nonempty, and so is $\mathrm{Proj}_n(p^*)$. Let $\tilde{\boldsymbol{v}}$ be a vertex of $\mathrm{Proj}_n(p^*)$. Then by definition, there exists a hyperplane $H$ of the form $H = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}^T\boldsymbol{x} = b\}$ for $\boldsymbol{a} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ with $a_n = 0$ and $b \in \mathbb{R}$ such that $H \cap \mathrm{Proj}_n(p^*) = \{\tilde{\boldsymbol{v}}\}$ and $\boldsymbol{a}^T\boldsymbol{x} \leq b$ for all $\boldsymbol{x} \in \mathrm{Proj}_n(p^*)$. Since $\tilde{\boldsymbol{v}} \in \mathrm{Proj}_n(p^*)$, there exists a number $t \in \mathbb{R}$ such that $\tilde{\boldsymbol{v}} + t\boldsymbol{e}_n \in \mathrm{Newt}(p^*)$. Among these numbers, let $t_1, t_2 \in \mathbb{R}$ be the minimum and maximum ones, respectively. Note that $t_1, t_2$ are not necessarily distinct. Let $\boldsymbol{u} = \tilde{\boldsymbol{v}} + t_1\boldsymbol{e}_n$ and $\boldsymbol{v} = \tilde{\boldsymbol{v}} + t_2\boldsymbol{e}_n$. Then the line segment $[\boldsymbol{u}, \boldsymbol{v}]$, possibly being a point when $t_1 = t_2$, is parallel to the $x_n$-axis and contained in $\mathrm{Newt}(p^*)$ by convexity.

Evidently, $\boldsymbol{a}^T\boldsymbol{\lambda}_m = \boldsymbol{a}^T\boldsymbol{e}_n = 0$. We claim that $[\boldsymbol{u}, \boldsymbol{v}]$ is the face of $\mathrm{Newt}(p^*)$ determined by the outward normal $\boldsymbol{a}$, which will complete the proof of the first assertion. In other words, we aim to prove that

$$H \cap \mathrm{Newt}(p^*) = [\boldsymbol{u}, \boldsymbol{v}] \quad \text{and} \quad \boldsymbol{a}^T\boldsymbol{x} \leq b \text{ for all } \boldsymbol{x} \in \mathrm{Newt}(p^*).$$

Let $\boldsymbol{x} \in \mathrm{Newt}(p^*)$ and $\tilde{\boldsymbol{x}} = (x_1, \ldots, x_{n-1}, 0)$. Then $\boldsymbol{a}^T\boldsymbol{x} = \boldsymbol{a}^T\tilde{\boldsymbol{x}} \leq b$ as $a_n = 0$ and $\tilde{\boldsymbol{x}} \in \mathrm{Proj}_n(p^*)$. To see the inclusion $H \cap \mathrm{Newt}(p^*) \subset [\boldsymbol{u}, \boldsymbol{v}]$, we further assume that $\boldsymbol{x} \in H \cap \mathrm{Newt}(p^*)$. Thus $\tilde{\boldsymbol{x}} \in H \cap \mathrm{Proj}_n(p^*) = \{\tilde{\boldsymbol{v}}\}$. This means that $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{v}}$. By the minimality of $t_1$ and maximality of $t_2$, we know that $\boldsymbol{x} \in [\boldsymbol{u}, \boldsymbol{v}]$. The opposite direction $H \cap \mathrm{Newt}(p^*) \supset [\boldsymbol{u}, \boldsymbol{v}]$ is clear from definition.

Moreover, assume that $\mathrm{Newt}(p)$ is not a line segment. Then $\mathrm{Newt}(p^*)$ cannot be a point or a line segment parallel to $\mathrm{Newt}(P_m(\boldsymbol{x}^{\boldsymbol{\lambda}_m}))$ by (3.2) and Lemma 2.2. This implies that $\mathrm{Proj}_n(p^*)$ has at least two different vertices. Taking another vertex of $\mathrm{Proj}_n(p^*)$ distinct from $\tilde{\boldsymbol{v}}$, and arguing along similar lines as above yields another edge of $\mathrm{Newt}(p)$ which has the direction vector $\boldsymbol{\lambda}_m$. The lemma therefore follows. □

From the above lemma, one sees that the direction vectors of edges of $\mathrm{Newt}(p)$ exhaust all possible choices of $q$-integer linear types. When $\mathrm{Newt}(p)$ is not a line segment, one can restrict attention to those vectors with multiple occurrences. Note that in our application, the Newton polytope of a given polynomial will be described by the set of its edges. Such a set can be easily deduced from the face lattice or the vertex-facet incidence matrix of the given Newton polytope, for which algorithms from computational geometry are well developed; see (Goodman et al., 2018, Chapter 26) and the references therein.

Given a set of points with cardinality $s \in \mathbb{N}$, it is known that the number of edges of the convex hull of this set is bounded by $\binom{s}{2}$ (cf. (Grünbaum, 2003, Theorem 2, Page 194)). Thus

Lemma 3.2 might offer us a superset of $q$-integer linear types of cardinality $\mathrm{O}(s^2)$ in the worst case. The following lemma, however, helps us bring it down to $\mathrm{O}(s)$.

**Lemma 3.3.** *With the assumptions of Lemma 3.2, for any $i \in \mathbb{N}$ with $1 \leq i \leq m$ and for any $\boldsymbol{j} \in \operatorname{supp}(p)$, there exists another vector $\tilde{\boldsymbol{j}} \in \operatorname{supp}(p)$ such that the line segment $[\boldsymbol{j}, \tilde{\boldsymbol{j}}]$ has the direction vector $\boldsymbol{\lambda}_i$, or equivalently, $\boldsymbol{j} - \tilde{\boldsymbol{j}} = k\boldsymbol{\lambda}_i$ for some nonzero integer $k$.*

*Proof.* There is nothing to show when $m = 0$, so assume that $m > 0$. By symmetry, it suffices to show that the assertion holds for $i = m$.

Again, we take $p^* = \boldsymbol{x}^{\boldsymbol{\alpha}} P_0 \prod_{i=1}^{m-1} P_i(\boldsymbol{x}^{\boldsymbol{\lambda}_i})$ and derive the decomposition (3.2) of $p$. Notice that $\operatorname{supp}(p)$ is nonempty as $p \neq 0$. Let $\boldsymbol{j} \in \operatorname{supp}(p)$. It follows from (3.2) that there is $\boldsymbol{j}^* \in \operatorname{supp}(p^*)$ and $k^* \in \operatorname{supp}(P_m)$ such that $\boldsymbol{j} = \boldsymbol{j}^* + k^* \boldsymbol{\lambda}_m$. Now consider the set

$$S = \{\bar{\boldsymbol{j}} \in \operatorname{supp}(p^*) \mid \bar{\boldsymbol{j}} = \boldsymbol{j}^* + k\boldsymbol{\lambda}_m \text{ for some } k \in \mathbb{Z}\}.$$

Then there exist $p_1^*, p_2^* \in \mathsf{R}[\boldsymbol{x}]$ with $\operatorname{supp}(p_1^*) = S$ and $\operatorname{supp}(p_2^*) = \operatorname{supp}(p^*) \setminus S$ such that $p^* = p_1^* + p_2^*$. It is evident that $\boldsymbol{j}^* \in S$. Thus $S$ is nonempty and then $p_1^*$ is nonzero. Let $\boldsymbol{\alpha}^* \in S$ be such that any element of $S$ can be written as $\boldsymbol{\alpha}^* + k\boldsymbol{\lambda}_m$ for some $k \in \mathbb{N}$, or equivalently, any monomial present in $p_1^*$ takes the form $\boldsymbol{x}^{\boldsymbol{\alpha}^* + k\boldsymbol{\lambda}_m}$ for some $k \in \mathbb{N}$. It then follows that there exists a nonzero univariate polynomial $P^* \in \mathsf{R}[y]$ such that $p_1^* = \boldsymbol{x}^{\boldsymbol{\alpha}^*} P^*(\boldsymbol{x}^{\boldsymbol{\lambda}_m})$.

On the other hand, by noticing that for any $\bar{\boldsymbol{j}} \in \operatorname{supp}(p_2^*) = \operatorname{supp}(p^*) \setminus S$, we have $\bar{\boldsymbol{j}} \neq \boldsymbol{j}^* + k\boldsymbol{\lambda}_m$ for all $k \in \mathbb{Z}$. Hence, $p$ can be decomposed as $p = f + g$, where $f = p_1^* P_m(\boldsymbol{x}^{\boldsymbol{\lambda}_m})$ and $g = p_2^* P_m(\boldsymbol{x}^{\boldsymbol{\lambda}_m})$ with $\operatorname{supp}(f) \cap \operatorname{supp}(g) = \emptyset$. As a consequence, $\operatorname{supp}(p) = \operatorname{supp}(f) \uplus \operatorname{supp}(g)$. Since $\boldsymbol{j} = \boldsymbol{j}^* + k^* \boldsymbol{\lambda}_m$, we have $\boldsymbol{j} \in \operatorname{supp}(f)$. Notice that $p_1^* = \boldsymbol{x}^{\boldsymbol{\alpha}^*} P^*(\boldsymbol{x}^{\boldsymbol{\lambda}_m})$. So $f = \boldsymbol{x}^{\boldsymbol{\alpha}^*} \tilde{P}(\boldsymbol{x}^{\boldsymbol{\lambda}_m})$ with $\tilde{P} = P^* P_m \in \mathsf{R}[y] \setminus \{0\}$. Then there exists $k \in \operatorname{supp}(\tilde{P})$ such that $\boldsymbol{j} = \boldsymbol{\alpha}^* + k\boldsymbol{\lambda}_m$. Since $P_m$ is $q$-primitive and of positive total degree, it possesses more than one monomial, and hence so does $\tilde{P}$. This implies that there is another element $\tilde{k} \in \operatorname{supp}(\tilde{P})$ distinct from $k$. Let $\tilde{\boldsymbol{j}} = \boldsymbol{\alpha}^* + \tilde{k}\boldsymbol{\lambda}_m$. Then $\tilde{\boldsymbol{j}} \in \operatorname{supp}(f) \subset \operatorname{supp}(p)$ and $\boldsymbol{j} - \tilde{\boldsymbol{j}} = (k - \tilde{k})\boldsymbol{\lambda}_m$. This concludes the proof. □

Combining Lemmas 3.2 and 3.3 suggests a simple geometric way to find candidates for all $q$-integer linear types of a given polynomial.

**Proposition 3.4.** *With the assumptions of Lemma 3.2, let $\Lambda_1$ be the multiset of direction vectors of edges of* $\operatorname{Newt}(p)$ *having no zero coordinates. Let $\boldsymbol{v} \in \operatorname{supp}(p)$ be fixed and let $\Lambda_2$ be the set consisting of direction vectors of line segments connecting $\boldsymbol{v}$ and all other points in* $\operatorname{supp}(p)$ *which have no zero coordinates.*

*(1) If the cardinality of $\Lambda_1$ is one then $p$ is $q$-integer linear of type $\boldsymbol{\lambda} \in \Lambda_1$.*

*(2) Otherwise, let $\Lambda_1^*$ be the subset of $\Lambda_1$ composed of elements with multiple occurrences. Then the intersection $\Lambda_1^* \cap \Lambda_2$ constitutes a superset of $q$-integer linear types of $p$. Moreover, with $s \in \mathbb{N}$ denoting the cardinality of* $\operatorname{supp}(p)$*, this superset has no more than $s - 1$ elements in total.*

Let $p$ be as given in Lemma 3.2 and assume further that $p$ is $q$-integer linear. Then one sees from the decomposition (3.1) and Lemma 2.2 that $\operatorname{Newt}(p)$ is the Minkowski sum of finitely many line segments. Such a polytope is called a *zonotope* in the literature. Zonotopes form an especially interesting and important class of convex polytopes; we refer to (Ziegler, 1995, Lecture 7) for more information. One of the key features of the zonotope $\operatorname{Newt}(p)$ is that the direction vectors of its edges are exactly those of its zones (namely the line segments present in the Minkowski sum), which, in our context, are all $q$-integer linear types $\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_m$ from (3.1). We therefore obtain the following necessary condition for a polynomial to be $q$-integer linear.

**Proposition 3.5.** *With the assumptions of Lemma 3.2, further assume that $p$ is $q$-integer linear. Then* $\operatorname{Newt}(p)$ *is a zonotope and none of the direction vectors of edges of* $\operatorname{Newt}(p)$ *has zero coordinates. As a consequence, for any integer $i$ with $1 \le i \le n$, there exists a unique vector in* $\operatorname{supp}(p)$ *whose $i$-th coordinate takes extremum value.*

*Proof.* Notice that none of the $q$-integer linear types of $p$ has zero coordinates. The first assertion is thus a direct result of the discussion preceding the proposition. In terms of the second assertion, we only show the argument on minimality for $i = n$, that is, we will prove that there exists only one vector in $\operatorname{supp}(p)$ whose $i$-th coordinate attains minimum. The rest follows by symmetry.

We proceed to using proof by contradiction. Suppose that there are at least two vectors in $\operatorname{supp}(p)$ whose $n$-th coordinate is equal to $\min_{\boldsymbol{x}\in\operatorname{supp}(p)}\{x_n\}$. Let $\boldsymbol{a} \in \operatorname{supp}(p)$ be one of these vectors. We claim that $H := \{\boldsymbol{x} \in \mathbb{R}^n \mid -x_n = -a_n\}$ is a supporting hyperplane of $\operatorname{Newt}(p)$. By the minimality of $a_n$, we know that $-x_n \le -a_n$ for all $\boldsymbol{x} \in \operatorname{supp}(p)$. It then follows from the convexity of $\operatorname{Newt}(p)$ that $-x_n \le -a_n$ for all $\boldsymbol{x} \in \operatorname{Newt}(p)$. Since $\boldsymbol{a} \in H \cap \operatorname{Newt}(p) \neq \emptyset$, the claim holds.

Let $F = H \cap \operatorname{Newt}(p)$. Then $F$ is a face of $\operatorname{Newt}(p)$ by the claim and thus is itself a Newton polytope by (Ziegler, 1995, Proposition 2.3(i)). By assumption, $F$ has at least two points and then possesses an edge, say $[\boldsymbol{u}, \boldsymbol{v}]$ for $\boldsymbol{u}, \boldsymbol{v} \in \operatorname{supp}(p)$. By (Ziegler, 1995, Proposition 2.3 (iii)), $[\boldsymbol{u}, \boldsymbol{v}]$ is also an edge of $\operatorname{Newt}(p)$, whose direction vector has zero $n$-th coordinate since $\boldsymbol{u}, \boldsymbol{v} \in F \subset H$, a contradiction with the first assertion. □

### 3.2. *Computation of univariate polynomials*

With candidates for the $q$-integer linear types at hand, we are able to find the corresponding univariate polynomials based on a $q$-counterpart of (Giesbrecht et al., 2019, Proposition 3.2).

**Proposition 3.6.** *With the assumptions of Lemma 3.2, let $\boldsymbol{\lambda} \in \mathbb{Z}^n$ with* $\gcd(\lambda_1, \dots, \lambda_n) = 1$, *$\lambda_1, \dots, \lambda_{n-1}$ not all zero and $\lambda_n > 0$. Let $P^* \in \mathsf{R}[y]$ be the content with respect to $x_1, \dots, x_{n-1}$ of the numerator of $p(x_1^{\lambda_n}, \dots, x_{n-1}^{\lambda_n}, y x_1^{-\lambda_1} \cdots x_{n-1}^{-\lambda_{n-1}})$. If $P^* \notin \mathsf{R}$ then $\boldsymbol{\lambda}$ is a $q$-integer linear type of $p$ with corresponding univariate polynomial $P^*(y^{1/\lambda_n}) \in \mathsf{R}[y]$. Otherwise, $\boldsymbol{\lambda}$ is not a $q$-integer linear type of $p$.*

In order to prove the above proposition, we first need to introduce some basic notions and lemmas. In the sequel of this subsection, we let $\mathbb{K}$ denote the quotient field of $\mathsf{R}$ and consider polynomials in $x_n$ over the field $\mathbb{K}(x_1, \dots, x_{n-1})$, all of which form the ring $\mathbb{K}(x_1, \dots, x_{n-1})[x_n]$. It is convenient to extend the definition of content and primitive part to polynomials in this setting. Let $p \in \mathbb{K}(x_1, \dots, x_{n-1})[x_n]$ be of the form $\sum_{i=0}^{d}(a_i/b)x_n^i$ for $d \in \mathbb{N}$ and $a_i, b \in \mathsf{R}[x_1, \dots, x_{n-1}]$. Then the *content* $\operatorname{cont}_{x_n}(p)$ of $p$ with respect to $x_n$ is defined as $\gcd(a_0, \dots, a_d)/b$ and the corresponding *primitive part* $\operatorname{prim}_{x_n}(p) = p/\operatorname{cont}_{x_n}(p)$. Evidently, $\operatorname{prim}_{x_n}(p) \in \mathsf{R}[\boldsymbol{x}]$. The definition of leading coefficient and degree extends to polynomials in $\mathbb{K}[x_1, \dots, x_n]$ in a natural manner.

**Lemma 3.7.** *Let $P \in \mathsf{R}[y] \setminus \mathsf{R}$ with $P(0) \neq 0$ and let $\boldsymbol{\lambda} \in \mathbb{Z}^n$ with* $\gcd(\lambda_1, \dots, \lambda_n) = 1$, *$\lambda_1, \dots, \lambda_{n-1}$ not all zero and $\lambda_n > 0$. Then*

*(i) for any factor $f \in \mathbb{K}(x_1, \dots, x_{n-1})[x_n]$ of $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ which is monic and irreducible over $\mathbb{K}(x_1, \dots, x_{n-1})$, there exists $c \in \mathbb{K}$, $\alpha_1, \dots, \alpha_{n-1} \in \mathbb{Z}$ and a factor $g \in \mathsf{R}[y]$ of $P$ such that $f = c x_1^{\alpha_1} \cdots x_{n-1}^{\alpha_{n-1}} g(\boldsymbol{x}^{\boldsymbol{\lambda}})$. Moreover, $0 < \deg(g) = \deg_{x_n}(f)/\lambda_n$.*

*(ii) $P$ is irreducible over $\mathsf{R}$ if and only if $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ is irreducible over $\mathbb{K}(x_1, \ldots, x_{n-1})$ if and only if $\operatorname{prim}_{x_n}(P(\boldsymbol{x}^{\boldsymbol{\lambda}}))$ is irreducible over $\mathsf{R}$.*

*Proof.* (i) Since $P(0) \neq 0$, all its roots in the algebraic closure $\overline{\mathbb{K}}$ of the field $\mathbb{K}$ are nonzero. In order to prove the assertion, it is sufficient to show that for any root $r \in \overline{\mathbb{K}}$ of $P$ , the polynomial $\boldsymbol{x}^{\boldsymbol{\lambda}} - r$ is irreducible over $\overline{\mathbb{K}}(x_1, \ldots, x_{n-1})$. For then, since $f \in \mathbb{K}(x_1, \ldots, x_{n-1})[x_n]$ is a monic and irreducible factor of $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$, it factors completely into irreducibles in $\overline{\mathbb{K}}(x_1, \ldots, x_{n-1})[x_n]$ as follows

$$f = \prod_{i=1}^{s} (x_1^{-\lambda_1} \cdots x_{n-1}^{-\lambda_{n-1}})(\boldsymbol{x}^{\boldsymbol{\lambda}} - r_i) = (x_1^{-\lambda_1} \cdots x_{n-1}^{-\lambda_{n-1}})^s \prod_{i=1}^{s} (\boldsymbol{x}^{\boldsymbol{\lambda}} - r_i),$$

where $s \in \mathbb{N}$ with $s \leq \deg(P)$ and the $r_i \in \overline{\mathbb{K}}$ are roots of $P$, and thus the assertion directly follows by letting $g = \operatorname{prim}_y(\prod_{i=1}^{s}(y - r_i))$.

Let $r \in \overline{\mathbb{K}}$ be a root of $P$ and suppose that $\boldsymbol{x}^{\boldsymbol{\lambda}} - r$ is reducible over $\overline{\mathbb{K}}(x_1, \ldots, x_{n-1})$. Then we have $\lambda_n > 1$. Consider the algebraic closure $\overline{\mathbb{K}(x_1, \ldots, x_{n-1})}$ of $\mathbb{K}(x_1, \ldots, x_{n-1})$ and let $\omega \in \overline{\mathbb{K}}$ be a $\lambda_n$-th root of unity so that $\omega^{\lambda_n} = 1$. Since $r$ is nonzero, the complete factorization of $\boldsymbol{x}^{\boldsymbol{\lambda}} - r$ over $\overline{\mathbb{K}(x_1, \ldots, x_{n-1})}$ is given by

$$\boldsymbol{x}^{\boldsymbol{\lambda}} - r = x_1^{\lambda_1} \cdots x_{n-1}^{\lambda_{n-1}} \prod_{i=0}^{\lambda_n - 1} \left( x_n - \omega^i r^{1/\lambda_n} x_1^{-\lambda_1/\lambda_n} \cdots x_{n-1}^{-\lambda_{n-1}/\lambda_n} \right).$$

It then follows from the reducibility of $\boldsymbol{x}^{\boldsymbol{\lambda}} - r$ over $\overline{\mathbb{K}}(x_1, \ldots, x_{n-1})$ that there exist $i_1, \ldots, i_k \in \{0, \ldots, \lambda_n - 1\}$ with $0 < k < \lambda_n$ such that

$$\prod_{j=1}^{k} \left( x_n - \omega^{i_j} r^{1/\lambda_n} x_1^{-\lambda_1/\lambda_n} \cdots x_{n-1}^{-\lambda_{n-1}/\lambda_n} \right) \in \overline{\mathbb{K}}(x_1, \ldots, x_{n-1})[x_n].$$

This implies that $(\lambda_i/\lambda_n)k \in \mathbb{Z}$ for all $i = 1, \ldots, n-1$. Thus $\lambda_n$ divides $k \cdot \gcd(\lambda_1, \ldots, \lambda_{n-1})$ in $\mathbb{Z}$. Since $\lambda_1, \ldots, \lambda_{n-1}$ are not all zero, $\gcd(\lambda_1, \ldots, \lambda_n) = 1$ and $\lambda_n > 1$, we have $\lambda_n$ divides $k$ in $\mathbb{Z}$, a contradiction since $0 < k < \lambda_n$.

(ii) For the first equivalence, the sufficiency is evident. In order to show the necessity, suppose that $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ is reducible over $\mathbb{K}(x_1, \ldots, x_{n-1})$. Let $f \in \mathbb{K}(x_1, \ldots, x_{n-1})[x_n]$ be an irreducible factor of $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$. Then the degree of $f$ in $x_n$ is less than $\lambda_n \deg(P)$. By assertion (i), we obtain that there exists a nontrivial factor $g \in \mathsf{R}[y]$ dividing $P$ in $\mathsf{R}[y]$ and $\deg(g) = \deg_{x_n}(f)/\lambda_n < \deg(P)$, a contradiction with the assumption that $P$ is irreducible over $\mathsf{R}$. Therefore, $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ is irreducible over $\mathbb{K}(x_1, \ldots, x_{n-1})$.

For the second equivalence, by Gauß' lemma, one easily sees that $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ is irreducible over $\mathbb{K}(x_1, \ldots, x_{n-1})$ if and only if $\operatorname{prim}_{x_n}(P(\boldsymbol{x}^{\boldsymbol{\lambda}}))$ is irreducible over $\mathsf{R}[x_1, \ldots, x_{n-1}]$. It thus amounts to showing the equivalence between the irreducibility of $\operatorname{prim}_{x_n}(P(\boldsymbol{x}^{\boldsymbol{\lambda}}))$ over $\mathsf{R}[x_1, \ldots, x_{n-1}]$ and its irreducibility over $\mathsf{R}$. The direction from $\mathsf{R}$ to $\mathsf{R}[x_1, \ldots, x_{n-1}]$ is trivial. In order to see the converse, notice that any nontrivial factor of $\operatorname{prim}_{x_n}(P(\boldsymbol{x}^{\boldsymbol{\lambda}}))$ can only belong to $\mathsf{R}[x_1, \ldots, x_{n-1}]$ since $\operatorname{prim}_{x_n}(P(\boldsymbol{x}^{\boldsymbol{\lambda}}))$ is irreducible over $\mathsf{R}[x_1, \ldots, x_{n-1}]$. On the other hand, the existence of any such a nontrivial factor would contradict with the fact that $\operatorname{prim}_{x_n}(P(\boldsymbol{x}^{\boldsymbol{\lambda}}))$ is primitive with respect to $x_n$. Accordingly, $\operatorname{prim}_{x_n}(P(\boldsymbol{x}^{\boldsymbol{\lambda}}))$ must be irreducible over $\mathsf{R}$. □

**Lemma 3.8.** *Let $p \in \mathsf{R}[\boldsymbol{x}]$ and $\boldsymbol{\lambda} \in \mathbb{Z}^n$ with $\gcd(\lambda_1, \ldots, \lambda_n) = 1$, $\lambda_1, \ldots, \lambda_{n-1}$ not all zero and $\lambda_n > 0$. Let $P \in \mathbb{K}[y]$ be such that $P(0) \neq 0$ and $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ divides $p$ in $\mathbb{K}(x_1, \ldots, x_{n-1})[x_n]$. Then $\boldsymbol{\lambda}$ is a q-integer linear type of $p$ with the corresponding univariate polynomial divided by $P$ in $\mathbb{K}[y]$.*

*Proof.* Let $f \in \mathsf{R}[y]$ be a primitive irreducible factor of $P$. Since $P(0) \neq 0$, then $f$ is $q$-primitive. Notice that $\boldsymbol{\lambda} \in \mathbb{Z}^n$ and $\lambda_n > 0$. So $\mathrm{prim}_{x_n}(f(\boldsymbol{x}^{\boldsymbol{\lambda}})) = \boldsymbol{x}^{\boldsymbol{\alpha}} f(\boldsymbol{x}^{\boldsymbol{\lambda}})$ for some $\boldsymbol{\alpha} \in \mathbb{N}^n$ with $\alpha_n = 0$. This implies that $\mathrm{prim}_{x_n}(f(\boldsymbol{x}^{\boldsymbol{\lambda}}))$ is a $q$-integer linear polynomial in $\mathsf{R}[\boldsymbol{x}]$ of type $\boldsymbol{\lambda}$. Because $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ divides $p$ in $\mathbb{K}(x_1, \ldots, x_{n-1})[x_n]$, so does $f(\boldsymbol{x}^{\boldsymbol{\lambda}})$. One then concludes from Lemma 3.7 (ii) that $\mathrm{prim}_{x_n}(f(\boldsymbol{x}^{\boldsymbol{\lambda}}))$ is an irreducible factor of $p$ over $\mathsf{R}$. Therefore, by Definition 3.1, $\boldsymbol{\lambda}$ is a $q$-integer linear type of $p$ and $f$ divides its corresponding polynomial in $\mathsf{R}[y]$. Since $f$ is arbitrary, the lemma follows. $\square$

We are now ready to prove Proposition 3.6.

*Proof of Proposition 3.6.* Assume that $P^* \in \mathsf{R}[y] \setminus \mathsf{R}$ and let $f \in \mathbb{K}[y]$ be a monic irreducible factor of $P^*$. Then $f(x_n)$ divides $p(x_1^{\lambda_n}, \ldots, x_{n-1}^{\lambda_n}, x_n x_1^{-\lambda_1} \cdots x_{n-1}^{-\lambda_{n-1}})$ in $\mathbb{K}(x_1, \ldots, x_{n-1})[x_n]$. Subsequently substituting $x_n$ by $x_n x_1^{\lambda_1} \cdots x_{n-1}^{\lambda_{n-1}}$ and then $x_i$ by $x_i^{1/\lambda_n}$ for $i = 1, \ldots, n-1$ yields that $f(x_n x_1^{\lambda_1/\lambda_n} \cdots x_{n-1}^{\lambda_{n-1}/\lambda_n})$ divides $p$ in $\overline{\mathbb{K}(x_1, \ldots, x_{n-1})}[x_n]$ where $\overline{\mathbb{K}(x_1, \ldots, x_{n-1})}$ denotes the algebraic closure of the field $\mathbb{K}(x_1, \ldots, x_{n-1})$. This implies that $f(y) \neq y$, for, otherwise, we would have that $x_n$ divides $p$ in $\overline{\mathbb{K}(x_1, \ldots, x_{n-1})}[x_n]$ and then $p(x_1, \ldots, x_{n-1}, 0) = 0$, a contradiction with the primitivity of $p$ with respect to $x_1$. Let $r \in \overline{\mathbb{K}}$ be a root of $f$. Then $r \neq 0$ and $f$ is its minimal polynomial in $\mathbb{K}[y]$. It follows from the divisibility of $p$ by $f(x_n x_1^{\lambda_1/\lambda_n} \cdots x_{n-1}^{\lambda_{n-1}/\lambda_n})$ that $p(x_1, \ldots, x_{n-1}, r x_1^{-\lambda_1/\lambda_n} \cdots x_n^{-\lambda_{n-1}/\lambda_n}) = 0$. Now consider the minimal polynomial $P \in \mathbb{K}[y]$ of $r^{\lambda_n}$. By Lemma 3.7 (ii), $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ is irreducible over $\mathbb{K}(x_1, \ldots, x_{n-1})$. Thus $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$, upon making it monic with respect to $x_n$, gives rise to the minimal polynomial of $x_n = r x_1^{-\lambda_1/\lambda_n} \cdots x_n^{-\lambda_{n-1}/\lambda_n}$. Therefore, $P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ divides $p$ in $\mathbb{K}(x_1, \ldots, x_{n-1})[x_n]$. One thus concludes from Lemma 3.8 that $\boldsymbol{\lambda}$ is a $q$-integer linear type of $p$, say $\boldsymbol{\lambda} = \boldsymbol{\lambda}_i$ for some integer $i$ with $1 \leq i \leq m$, and then $P$ divides $P_i$ in $\mathbb{K}[y]$. Notice that $f$ is the minimal polynomial of $r$ and $P(r^{\lambda_n}) = 0$. So $f$ divides $P(y^{\lambda_n})$ and then $P_i(y^{\lambda_n})$ in $\mathbb{K}[y]$. As $f$ is arbitrary, we have that $P^*$ divides $P_i(y^{\lambda_n})$ in $\mathbb{K}[y]$. Since both polynomials are $q$-primitive and $\lambda_n > 0$, then $P^*$ divides $P_i(y^{\lambda_n})$ in $\mathsf{R}[y]$ by Gauß' lemma.

In order to show the first assertion, it remains to verify that $P_i(y^{\lambda_n})$ divides $P^*$ in $\mathsf{R}[y]$, and then $P^*$ and $P_i(y^{\lambda_n})$ only differ by a unit in $\mathsf{R}$, yielding the assertion.

Since $\boldsymbol{\lambda} = \boldsymbol{\lambda}_i$, by a simple calculation, one sees from (3.1) that $P_i(y^{\lambda_n})$ divides all coefficients of $p(x_1^{\lambda_n}, \ldots, x_{n-1}^{\lambda_n}, y x_1^{-\lambda_1} \cdots x_{n-1}^{-\lambda_{n-1}})$ with respect to $x_1, \ldots, x_{n-1}$. By the definition of $P^*$, we obtain that $P_i(y^{\lambda_n})$ divides $P^*$ in $\mathsf{R}[y]$. This actually also shows that $P^* \notin \mathsf{R}$ if $\boldsymbol{\lambda} = \boldsymbol{\lambda}_i$ is a $q$-integer linear type of $p$, because $P_i \notin \mathsf{R}$. This completes the proof of the second assertion. $\square$

### 3.3. *Algorithm and example*

Assembling everything together yields our first approach.

**MultivariateQILD$_1$.** Given a polynomial $p \in \mathsf{R}[\boldsymbol{x}]$, compute its $q$-integer linear decomposition.

1. If $p \in \mathsf{R}$ then set $c = p$; and return $c$.
2. Set $c = \mathrm{cont}(p)$ and $f = \mathrm{prim}(p)$. If $\mathrm{supp}(f)$ is a singleton then set $\boldsymbol{\alpha}$ to be the only element and update $c = cf/\boldsymbol{x}^{\boldsymbol{\alpha}}$; and return $c\boldsymbol{x}^{\boldsymbol{\alpha}}$.
3. If $n = 1$ then set $\alpha_1$ to be the lowest degree of $f$ with respect to $x_1$, $m = 1$, $\lambda_{m1} = 1$ and $P_m(y) = f(y)/y^{\alpha_1}$; and return $c\, x_1^{\alpha_1} \prod_{i=1}^m P_i(x_1^{\lambda_{i1}})$.
4. Set $\boldsymbol{\alpha} = \boldsymbol{0}$, $P_0 = 1$, $m = 0$.

   For $i = 1, \ldots, n$ do

   4.1 Set $g = \mathrm{cont}_{x_i}(f)$, and update $f = \mathrm{prim}_{x_i}(f)$.

4.2 If $g \neq 1$ then call the algorithm recursively with input $g \in \mathsf{R}[x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$, returning

$$g = x_1^{\tilde{\alpha}_1} \cdots x_{i-1}^{\tilde{\alpha}_{i-1}} x_{i+1}^{\tilde{\alpha}_{i+1}} \cdots x_n^{\tilde{\alpha}_n} \tilde{P}_0 \prod_{j=1}^{\tilde{m}} \tilde{P}_j(x_1^{\tilde{\lambda}_{j1}} \cdots x_{i-1}^{\tilde{\lambda}_{j,i-1}} x_{i+1}^{\tilde{\lambda}_{j,i+1}} \cdots x_n^{\tilde{\lambda}_{jn}}),$$

update $\boldsymbol{\alpha} = \boldsymbol{\alpha} + (\tilde{\alpha}_1, \ldots, \tilde{\alpha}_{i-1}, 0, \tilde{\alpha}_{i+1}, \ldots, \tilde{\alpha}_n)$, $P_0 = P_0\tilde{P}_0$, and for $j = 1, \ldots, \tilde{m}$ iteratively update $m = m + 1$, $\boldsymbol{\lambda}_m = (\tilde{\lambda}_{j1}, \ldots, \tilde{\lambda}_{j,i-1}, 0, \tilde{\lambda}_{j,i+1}, \ldots, \tilde{\lambda}_{jn})$, $P_m(y) = \tilde{P}_j(y)$.

5. If $\deg(f) = 0$ then update $c = cf$; and return $c\,\boldsymbol{x}^{\boldsymbol{\alpha}} P_0 \prod_{i=1}^{m} P_i(\boldsymbol{x}^{\boldsymbol{\lambda}_i})$.
6. Find the multiset $\Lambda$ of direction vectors of edges of $\mathrm{Newt}(f)$ having no zero coordinates.
7. If $\Lambda$ has more than one element then
   7.1 Update $\Lambda$ to be its subset composed of elements with multiple occurrences.
   7.2 For fixed $\boldsymbol{v} \in \mathrm{supp}(f)$, find the set $\tilde{\Lambda}$ consisting of direction vectors of line segments connecting $\boldsymbol{v}$ and all other points in $\mathrm{supp}(p)$ which have no zero coordinates.
   7.3 Update $\Lambda$ to be $\Lambda \cap \tilde{\Lambda}$.
8. For $\boldsymbol{\lambda}$ in $\Lambda$ do
   8.1 Set $P^*(y)$ to be the content of the numerator of $f(x_1^{\lambda_n}, \ldots, x_{n-1}^{\lambda_n}, y x_1^{-\lambda_1} \cdots x_{n-1}^{-\lambda_{n-1}})$ with respect to $x_1, \ldots, x_{n-1}$.
   8.2 If $\deg(P^*) > 0$ then

       Update $m = m + 1$, $\boldsymbol{\lambda}_m = \boldsymbol{\lambda}$, $P_m(y) = P^*(y^{1/\lambda_n})$.
       Set $f^*, g^* \in \mathsf{R}[x_1, \ldots, x_n]$ to be the numerator and denominator of $P_m(\boldsymbol{x}^{\boldsymbol{\lambda}})$, and update $f = f/f^*$ and $\alpha_i = \alpha_i + \deg_{x_i}(g^*)$ for $i = 1, \ldots, n-1$.
9. If $\deg(f) > 0$ then update $P_0 = P_0 f$ else update $c = cf$.
10. Return $c\,\boldsymbol{x}^{\boldsymbol{\alpha}} P_0 \prod_{i=1}^{m} P_i(\boldsymbol{x}^{\boldsymbol{\lambda}_i})$.

**Theorem 3.9.** *Let $p \in \mathsf{R}[\boldsymbol{x}]$. Then the algorithm* **MultivariateQILD**$_1$ *terminates and correctly computes the q-integer linear decomposition of p.*

*Proof.* This is evident by Propositions 3.4 and 3.6. □

**Remark 3.10.** *If one is merely interested in only determining the q-integer linearity of the input polynomial $p \in \mathsf{R}[\boldsymbol{x}]$, rather than the full q-integer linear decomposition, then the above algorithm can be easily modified: any of the following conditions will trigger the adapted algorithm to terminate early, returning that p is not q-integer linear.*

- *In Step 4.2, the polynomial g turns out to be non-q-integer linear.*
- *(Proposition 3.5) In Step 6, the Newton polytope of f is not a zonotope; or there exists an edge of* $\mathrm{Newt}(f)$ *whose direction vector has zero coordinates. In particular, the support* $\mathrm{supp}(f)$ *has more than one element whose certain coordinate attains the extremum value.*
- *(Proposition 3.6) In Step 8.2, the case of* $\deg(P^*) = 0$ *happens, that is, the candidate $\boldsymbol{\lambda}$ currently under investigation is fake.*
- *(Definition 3.1) In Step 10, we have* $\deg(P_0) > 0$.

**Example 3.11.** *Consider the polynomial* $p \in \mathbb{Z}[q, q^{-1}][x_1, x_2, x_3, x_4]$ *of the form*

$$\begin{aligned} p = {} & 2q^2x_1^9x_2^{12}x_3^{13} + 2qx_1^8x_2^{14}x_3^{13} + 2qx_1^8x_2^{14}x_3^{12}x_4 + 18q^2x_1^{11}x_2^8x_3^{16}x_4^5 + 18qx_1^{10}x_2^{10}x_3^{16}x_4^5 \\ & + 18qx_1^{10}x_2^{10}x_3^{15}x_4^6 - 2qx_1^5x_2^{20}x_3^7x_4^7 - 2x_1^4x_2^{22}x_3^7x_4^7 - 2x_1^4x_2^{22}x_3^6x_4^8 - 18qx_1^7x_2^{16}x_3^{10}x_4^{12} \\ & - 18x_1^6x_2^{18}x_3^{10}x_4^{12} - 18x_1^6x_2^{18}x_3^9x_4^{13} + 7q^2x_1x_2^{28}x_3x_4^{14} + 7qx_2^{30}x_3x_4^{14} + 7qx_2^{30}x_4^{15} \\ & + 6q^4x_1^{15}x_3^{22}x_4^{15} + 6q^3x_1^{14}x_2^2x_3^{22}x_4^{15} + 6q^3x_1^{14}x_2^2x_3^{21}x_4^{16} + 63q^2x_1^3x_2^{24}x_3^4x_4^{19} \\ & + 63qx_1^2x_2^{26}x_3^4x_4^{19} + 63qx_1^2x_2^{26}x_3^3x_4^{20} - 6q^3x_1^{11}x_2^8x_3^{16}x_4^{22} - 6q^2x_1^{10}x_2^{10}x_3^{16}x_4^{22} \\ & - 6q^2x_1^{10}x_2^{10}x_3^{15}x_4^{23} + 21q^4x_1^7x_2^{16}x_3^{10}x_4^{29} + 21q^3x_1^6x_2^{18}x_3^{10}x_4^{29} + 21q^3x_1^6x_2^{18}x_3^9x_4^{30} \end{aligned} \tag{3.3}$$

*In order to compute the q-integer linear decomposition of the polynomial p over* $\mathbb{Z}[q, q^{-1}]$*, the algorithm* **MultivariateQILD**$_1$ *first tries to find candidates for all possible q-integer linear types of p. In this respect, it computes the Newton polytope of p from its support* $\mathrm{supp}(p)$*, which can be readily read out from* (3.3)*, and finds that* $\mathrm{Newt}(p)$ *possesses 11 vertices:*

$$\begin{aligned} \Big\{ & v_0 := (9,12,13,0), v_1 := (8,14,13,0), v_2 := (8,14,12,1), v_3 := (1,28,1,14), \\ & v_4 := (0,30,1,14), v_5 := (0,30,0,15), v_6 := (15,0,22,15), v_7 := (14,2,22,15), \\ & v_8 := (7,16,10,29), v_9 := (6,18,10,29), v_{10} := (6,18,9,30) \Big\}, \end{aligned}$$

*and 19 edges:*

$$\begin{aligned} \Big\{ & [v_1,v_4], [v_4,v_9], [v_7,v_9], [v_1,v_7], [v_4,v_5], [v_1,v_2], [v_2,v_5], [v_5,v_{10}], [v_9,v_{10}], [v_0,v_2], \\ & [v_0,v_1], [v_6,v_7], [v_6,v_8], [v_8,v_9], [v_0,v_3], [v_3,v_5], [v_0,v_6], [v_3,v_8], [v_8,v_{10}] \Big\}. \end{aligned}$$

*Based on Proposition 3.4 (namely Steps 6-7), one obtains three candidates for q-integer linear types of p, that is,* $(-1,2,-1,1)$*,* $(2,-4,3,5)$*,* $(-4,8,6,7)$*. A subsequent content computation for each candidate finally leads to the following q-integer linear decomposition*

$$p = x_1^8x_2^{12}x_3^{12} \cdot P_0 \cdot P_1(x_1^2x_2^{-4}x_3^3x_4^5) \cdot P_2(x_1^{-4}x_2^8x_3^{-6}x_4^7), \tag{3.4}$$

*where* $P_0 = qx_1x_3 + x_2^2x_3 + x_2^2x_4$*,* $P_1(y) = 3q^2y^3 + qy + 1$ *and* $P_2(y) = 7qy^2 - 2y + 2q$*.*

*Notice that there are two elements in the support* $\mathrm{supp}(p)$ *(namely the exponent vectors of the first two monomials in* (3.3)*) attaining the minimum value of* $x_4$*. One thus immediately sees from Proposition 3.5 that the given polynomial p is not q-integer linear. Also, the candidate* $(-1,2,-1,1)$ *turns out to be fake, implying, once again, the non-q-integer linearity of p.*

## 4. $q$-Integer linear decomposition: the second approach

In this section we present our second approach for computing the $q$-integer linear decomposition of a polynomial in an arbitrary number of variables. This approach uses a bivariate-based scheme, where the base bivariate case is tackled by the first approach from the proceeding section. In order to describe it concisely, we need a $q$-analogue of (Abramov and Petkovšek, 2002, Proposition 7). To this end, we require two technical lemmas. The first one corresponds to (Abramov and Petkovšek, 2002, Lemma 2) but restricted to the case of Laurent polynomials.

**Lemma 4.1.** *Let* $p \in \mathsf{R}[x, x^{-1}]$ *be a nonzero Laurent polynomial. If there exists a nonzero integer a and a nonzero element* $c \in \mathsf{R}$ *such that* $p(q^a x) = cp(x)$*, then* $c = q^{am}$ *for some* $m \in \mathbb{Z}$ *and* $p(x)/x^m \in \mathsf{R}$*.*

*Proof.* The assertion is clear if $p$ has only one monomial. Otherwise, let $x^i$ and $x^j$ with $i, j \in \mathbb{Z}$ be two monomials of $p$. Extracting their coefficients in the identity $p(q^a x) = cp(x)$ gives $q^{ai} = c = q^{aj}$. Thus $c$ has the form $q^{ai}$ for some $i \in \mathbb{Z}$ and all the exponents $j$ of the monomials in $p$ satisfy $a(j - i) = 0$, yielding $j = i$ as $a$ is nonzero. The lemma follows. $\square$

Evidently, the above lemma remains valid by replacing the ring $\mathsf{R}$ with any of its ring extensions which is independent of the variable $x$, or changing the variable $x$ to any its rational power $x^r$ for $r \in \mathbb{Q}$. The next lemma plays the role of (Abramov and Petkovšek, 2002, Lemma 3) in the $q$-shift setting, which describes a nice structure of $q$-shift invariant bivariate polynomials.

**Lemma 4.2.** *Let $p \in \mathsf{R}[x, y]$. If there exists $c \in \mathsf{R}$ and $a, b \in \mathbb{Z}$, not both zero, such that $p(q^a x, q^b y) = cp(x, y)$, then there is a univariate polynomial $P \in \mathsf{R}[y]$ and four integers $\alpha, \beta, \lambda, \mu$ with $\lambda, \mu$ not both zero such that $p = x^\alpha y^\beta P(x^\lambda y^\mu)$.*

*Proof.* Without loss of generality, we assume that $a$ is nonzero. Otherwise, we can switch the roles of $x$ and $y$ in the following proof. Define $h(x, y) = p(x, yx^{b/a})$. Then $h \in \mathsf{R}[x^{1/a}, x^{-1/a}, y]$ and $p(x, y) = h(x, yx^{-b/a})$. Using $p(q^a x, q^b y) = cp(x, y)$, a simple calculation shows that $h(q^a x, y) = p(q^a x, q^b yx^{b/a}) = ch(x, y)$. Viewing $h$ as a Laurent polynomial in $x^{1/a}$ over $\mathsf{R}[y]$, Lemma 4.1 implies that $h/x^{m/a} \in \mathsf{R}[y]$ for some $m \in \mathbb{Z}$. From the definition of $h$ we have that $i+(b/a)j = m/a$ for all $(i, j) \in \operatorname{supp}(p)$. Let $x^\alpha y^\beta$ with $\alpha, \beta \in \mathbb{N}$ be the trailing monomial in $p$, and let $\lambda, \mu \in \mathbb{Z}$ be such that $\lambda/\mu = -b/a$, $\gcd(\lambda, \mu) = 1$ and $\mu > 0$. Then $\mu(i - \alpha) = \lambda(j - \beta)$ for all $(i, j) \in \operatorname{supp}(p)$. By the coprimeness of $\lambda$ and $\mu$, one obtains that for any $(i, j) \in \operatorname{supp}(p)$, there exists $k \in \mathbb{N}$ such that $(i, j) = (\alpha, \beta) + k(\lambda, \mu)$. It thus follows that $p = x^\alpha y^\beta P(x^\lambda y^\mu)$ for some $P \in \mathsf{R}[y]$. $\square$

From the above lemma, we are then able to establish the fact that the problem of multivariate $q$-integer linearity is made up of a collection of subproblems of bivariate $q$-integer linearity.

**Proposition 4.3.** *Let $p \in \mathsf{R}[\boldsymbol{x}]$. Then there exists a univariate polynomial $P \in \mathsf{R}[y]$ and two vectors $\boldsymbol{\alpha} \in \mathbb{N}^n$, $\boldsymbol{\lambda} \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}$ such that $p = \boldsymbol{x}^{\boldsymbol{\alpha}} P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ if and only if for each pair $(i, j)$ with $1 \le i < j \le n$, there is a polynomial $P_{ij}(y) \in \mathsf{R}[x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_{j-1}, x_{j+1}, \dots, x_n][y]$ and four integers $\beta_{ij}, \beta_{ji}, \mu_{ij}, \mu_{ji}$ with $\mu_{ij}, \mu_{ji}$ not both zero such that $p = x_i^{\beta_{ij}} x_j^{\beta_{ji}} P_{ij}(x_i^{\mu_{ij}} x_j^{\mu_{ji}})$.*

*Proof.* The necessity is clear. For the sufficiency, we proceed by induction on the number $n$ of variables. There is nothing to show in the base case where $n = 1$. Assume that $n > 1$ and the assertion holds for $n - 1$.

Consider $p$ as a polynomial in $x_1, \dots, x_{n-1}$ over $\mathsf{R}[x_n]$. By the induction hypothesis, there is a polynomial $P^* \in \mathsf{R}[x_n][y]$ and two vectors $(\alpha_1^*, \dots, \alpha_{n-1}^*) \in \mathbb{N}^{n-1}$, $(\lambda_1^*, \dots, \lambda_{n-1}^*) \in \mathbb{Z}^{n-1}$ with the $\lambda_i^*$ not all zero such that

$$p(x_n)(x_1, \dots, x_{n-1}) = x_1^{\alpha_1^*} \cdots x_{n-1}^{\alpha_{n-1}^*} P^*(x_1^{\lambda_1^*} \cdots x_{n-1}^{\lambda_{n-1}^*}).$$

We may assume without loss of generality that $\lambda_1^* \neq 0$. Regarding $P^*$ as an element of $\mathsf{R}[y, x_n]$, we rewrite the preceding equation as

$$p(x_1, \dots, x_n) = x_1^{\alpha_1^*} \cdots x_{n-1}^{\alpha_{n-1}^*} P^*(x_1^{\lambda_1^*} \cdots x_{n-1}^{\lambda_{n-1}^*}, x_n). \tag{4.1}$$

By taking $i = 1$ and $j = n$ in the assumption, we know that $p = x_1^{\beta_{1n}} x_n^{\beta_{n1}} P_{1n}(x_1^{\mu_{1n}} x_n^{\mu_{n1}})$ for $P_{1n} \in \mathsf{R}[x_2, \dots, x_{n-1}][y]$ and $\beta_{1n}, \beta_{n1}, \mu_{1n}, \mu_{n1} \in \mathbb{Z}$ with $\mu_{1n}, \mu_{n1}$ not both zero. Therefore,

$$p(q^{\mu_{n1}} x_1, x_2, \dots, x_{n-1}, q^{-\mu_{1n}} x_n) = cp(x_1, \dots, x_n) \quad \text{with } c = q^{\beta_{1n}\mu_{n1} - \beta_{n1}\mu_{1n}} \in \mathsf{R}.$$

It follows from (4.1) that $P^*(q^{\mu_{n1}\lambda_1^*}x_1^{\lambda_1^*}\cdots x_{n-1}^{\lambda_{n-1}^*}, q^{-\mu_{1n}}x_n) = cq^{-\mu_{n1}\alpha_1^*}P^*(x_1^{\lambda_1^*}\cdots x_{n-1}^{\lambda_{n-1}^*}, x_n)$, that is,

$$P^*(q^{\mu_{n1}\lambda_1^*}y, q^{-\mu_{1n}}x_n) = cq^{-\mu_{n1}\alpha_1^*}P^*(y, x_n).$$

Applying Lemma 4.2 to $P^*(y, x_n)$ yields that there is a univariate polynomial $P \in \mathsf{R}[y]$ and four integers $\alpha_n, \alpha_n^*, \lambda_n, \lambda_n^*$ with $\lambda_n, \lambda_n^*$ not both zero such that $P^*(y, x_n) = y^{\alpha_n^*}x_n^{\alpha_n}P(y^{\lambda_n^*}x_n^{\lambda_n})$. Substituting $y = x_1^{\lambda_1^*}\cdots x_{n-1}^{\lambda_{n-1}^*}$ into this equation, together with (4.1), implies that $p = \boldsymbol{x}^{\boldsymbol{\alpha}}P(\boldsymbol{x}^{\boldsymbol{\lambda}})$ with $\boldsymbol{\alpha} = (\alpha_1^* + \lambda_1^*\alpha_n^*, \ldots, \alpha_{n-1}^* + \lambda_{n-1}^*\alpha_n^*, \alpha_n)$ and $\boldsymbol{\lambda} = (\lambda_1^*\lambda_n^*, \ldots, \lambda_{n-1}^*\lambda_n^*, \lambda_n)$. The proof follows by noticing that $\boldsymbol{\lambda} \neq \mathbf{0}$. □

Inspired by the above proposition, we propose an algorithm which takes a multivariate polynomial as input and computes its $q$-integer linear decomposition in an iterative fashion. At each iteration step, only two variables are used with the others treated as coefficient parameters.

**MultivariateQILD$_2$.** Given a polynomial $p \in \mathsf{R}[\boldsymbol{x}]$, compute its $q$-integer linear decomposition.

1. If $p \in \mathsf{R}$ then set $c = p$; and return $c$.
2. Set $c = \mathrm{cont}(p)$ and $f = \mathrm{prim}(p)$. If $\mathrm{supp}(f)$ is a singleton then set $\boldsymbol{\alpha}$ to be the only element and update $c = cf/\boldsymbol{x}^{\boldsymbol{\alpha}}$; and return $c\boldsymbol{x}^{\boldsymbol{\alpha}}$.
3. If $n = 1$ then set $\alpha_1$ to be the lowest degree of $f$ with respect to $x_1$, $m = 1$, $\lambda_{m1} = 1$ and $P_m(y) = f(y)/y^{\alpha_1}$; and return $c\,x_1^{\alpha_1}\prod_{i=1}^m P_i(x_1^{\lambda_{i1}})$.
4. If $n = 2$ then call the algorithm **MultivariateQILD**$_1$ with input $f \in \mathsf{R}[x_1, x_2]$ to compute its $q$-integer linear decomposition

   $$f = x_1^{\alpha_1}x_2^{\alpha_2}P_0\prod_{i=1}^m P_i(x_1^{\lambda_{i1}}x_2^{\lambda_{i2}});$$

   and then return $c\,x_1^{\alpha_1}x_2^{\alpha_2}P_0\prod_{i=1}^m P_i(x_1^{\lambda_{i1}}x_2^{\lambda_{i2}})$.
5. Set $\boldsymbol{\alpha} = \mathbf{0}$, $P_0 = 1$, $m = 0$ and $g = \mathrm{cont}_{x_1,x_2}(f)$, and update $f = \mathrm{prim}_{x_1,x_2}(f)$.
6. If $g \neq 1$ then call the algorithm recursively with input $g \in \mathsf{R}[x_3, \ldots, x_n]$, returning

   $$g = x_3^{\tilde\alpha_3}\cdots x_n^{\tilde\alpha_n}\tilde P_0\prod_{i=1}^{\tilde m}\tilde P_i(x_3^{\tilde\lambda_{i3}}\cdots x_n^{\tilde\lambda_{in}}),$$

   update $\boldsymbol{\alpha} = \boldsymbol{\alpha} + (0, 0, \tilde\alpha_3, \ldots, \tilde\alpha_n)$, $P_0 = P_0\tilde P_0$, and for $i = 1, \ldots, \tilde m$ iteratively update $m = m+1$, $\boldsymbol{\lambda}_m = (0, 0, \tilde\lambda_{i3}, \ldots, \tilde\lambda_{in})$, $P_m(y) = \tilde P_i(y)$.
7. If $\mathrm{supp}(f)$ is a singleton then set $\boldsymbol{\alpha}^*$ to be the only element and update $\boldsymbol{\alpha} = \boldsymbol{\alpha} + \boldsymbol{\alpha}^*$, $c = cf/\boldsymbol{x}^{\boldsymbol{\alpha}^*}$; and return $c\,\boldsymbol{x}^{\boldsymbol{\alpha}}P_0\prod_{i=1}^m P_i(\boldsymbol{x}^{\boldsymbol{\lambda}_i})$.
8. Set $\Lambda_1 = \{((1), f(y, x_2, \ldots, x_n))\}$.
   For $k = 1, \ldots, n-1$ do
   - 8.1 Set $\Lambda_{k+1} = \{\}$.
   - 8.2 For $((\mu_1, \ldots, \mu_k), h(y, x_{k+1}, \ldots, x_n))$ in $\Lambda_k$ do

Call the algorithm **MultivariateQILD**$_1$ with input $h \in \mathsf{R}[x_{k+2}, \ldots, x_n][y, x_{k+1}]$ to compute its $q$-integer linear decomposition

$$h = y^{\alpha^*} x_{k+1}^{\beta^*} P_0^* \prod_{i=1}^{m^*} P_i^*(y^{\lambda_i^*} x_{k+1}^{\mu_i^*}, x_{k+2}, \ldots, x_n), \tag{4.2}$$

where $P_0^* \in \mathsf{R}[y, x_{k+1}, \ldots, x_n]$ and $P_i^*(y, x_{k+2}, \ldots, x_n) \in \mathsf{R}[y, x_{k+2}, \ldots, x_n]$; then update $\boldsymbol{\alpha}$ by adding the vector $(\mu_1\alpha^*, \ldots, \mu_k\alpha^*, \beta^*, 0, \ldots, 0)$, update $P_0$ by multiplying $P_0^*(x_1^{\mu_1} \cdots x_k^{\mu_k}, x_{k+1}, \ldots, x_n)$ and update $\Lambda_{k+1}$ by joining the elements $((\mu_1\lambda_i^*, \ldots, \mu_k\lambda_i^*, \mu_i^*), P_i^*(y, x_{k+2}, \ldots, x_n))$ for $i = 1, \ldots, m^*$.

9. Set $g \in \mathsf{R}[\boldsymbol{x}]$ to be the denominator of $P_0$. Update $P_0$ to be its numerator, update $\alpha_i = \alpha_i - \deg_{x_i}(g)$ for $i = 1, \ldots, n-1$, and for $(\boldsymbol{\mu}, h(y))$ in $\Lambda_n$ iteratively update $m = m + 1$, $\boldsymbol{\lambda}_m = \boldsymbol{\mu}$ and $P_m(y) = h(y)$.
10. Return $c\, \boldsymbol{x}^{\boldsymbol{\alpha}} P_0 \prod_{i=1}^m P_i(\boldsymbol{x}^{\boldsymbol{\lambda}_i})$.

**Theorem 4.4.** *Let $p \in \mathsf{R}[\boldsymbol{x}]$. Then the algorithm* **MultivariateQILD**$_2$ *correctly computes the $q$-integer linear decomposition of $p$.*

*Proof.* The correctness immediately follows from Proposition 4.3. □

**Example 4.5.** *Consider the same polynomial $p$ given by (3.3) as Example 3.11. In order to compute its $q$-integer linear decomposition over $\mathbb{Z}[q, q^{-1}]$, the algorithm* **MultivariateQILD**$_2$ *(mainly Step 8) proceeds in the following three stages with their respective Newton polytopes plotted in Figure 1. Firstly, by viewing $p$ as a polynomial in $x_1, x_2$ over $\mathbb{Z}[q, q^{-1}, x_3, x_4]$, applying the algorithm* **MultivariateQILD**$_1$ *to $p$ gives*

$$p = x_1^{15} P^{(1)}(x_1^{-1}x_2^2, x_3, x_4) \tag{4.3}$$

*with*

$$\begin{aligned}
P^{(1)}(y, x_3, x_4) = {} & 7qy^{15}x_3x_4^{14} + 7qy^{15}x_4^{15} + 7q^2y^{14}x_3x_4^{14} + 63qy^{13}x_3^4x_4^{19} + 63qy^{13}x_3^3x_4^{20} \\
& + 63q^2y^{12}x_3^4x_4^{19} - 2y^{11}x_3^7x_4^7 - 2y^{11}x_3^6x_4^8 - 2qy^{10}x_3^7x_4^7 + 21q^3y^9x_3^{10}x_4^{29} - 18y^9x_3^{10}x_4^{12} \\
& + 21q^3y^9x_3^9x_4^{30} - 18y^9x_3^9x_4^{13} + 21q^4y^8x_3^{10}x_4^{29} - 18qy^8x_3^{10}x_4^{12} + 2qy^7x_3^{13} + 2qy^7x_3^{12}x_4 \\
& + 2q^2y^6x_3^{13} - 6q^2y^5x_3^{16}x_4^{22} + 18qy^5x_3^{16}x_4^5 - 6q^2y^5x_3^{15}x_4^{23} + 18qy^5x_3^{15}x_4^6 - 6q^3y^4x_3^{16}x_4^{22} \\
& + 18q^2y^4x_3^{16}x_4^5 + 6q^3yx_3^{22}x_4^{15} + 6q^3yx_3^{21}x_4^{16} + 6q^4x_3^{22}x_4^{15}.
\end{aligned}$$

*There is only one $q$-integer linear type, namely $(-1, 2)$, of $p$ over $\mathbb{Z}[q, q^{-1}, x_3, x_4]$. Next, with input $P^{(1)}(y, x_3, x_4) \in \mathbb{Z}[q, q^{-1}, x_4][y, x_3]$, calling the algorithm* **MultivariateQILD**$_1$ *again and substituting $y = x_1^{-1}x_2^2$ yields*

$$p = x_2^{28} \cdot P_0 \cdot P^{(2)}(x_1^2x_2^{-4}x_3^3, x_4), \tag{4.4}$$

*where $P_0 = qx_1x_3 + x_2^2x_3 + x_2^2x_4$ and $P^{(2)}(y, x_4) = 6q^3y^7x_4^{15} - 6q^2y^5x_4^{22} + 18qy^5x_4^5 + 2qy^4 + 21q^3y^3x_4^{29} - 18y^3x_4^{12} - 2y^2x_4^7 + 63qyx_4^{19} + 7qx_4^{14}$. The vector $(2, -4, 3)$ is then the only $q$-integer linear type of $p$ over $\mathbb{Z}[q, q^{-1}, x_4]$. Finally, the last call to the algorithm* **MultivariateQILD**$_1$ *with input $P^{(2)}(y, x_4) \in \mathbb{Z}[q, q^{-1}][y, x_4]$, along with the substitution $y = x_1^2x_2^{-4}x_3^3$, leads to the*

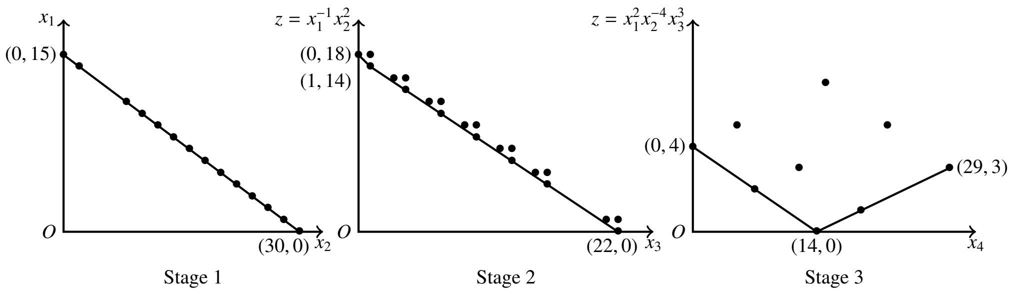


Figure 1: Newton polytopes constructed in the three stages in Example 4.5.

*desired decomposition* (3.4)*. The two q-integer linear types* $(2, -4, 3, 5)$ *and* $(-4, -8, -6, 7)$ *of* $p$ *over* $\mathbb{Z}[q, q^{-1}]$ *have been correctly recovered.*

*From* (4.3) *and* (4.4)*, one sees that* $p$ *is q-integer linear over* $\mathbb{Z}[q, q^{-1}, x_3, x_4]$ *but it is not q-integer linear over* $\mathbb{Z}[q, q^{-1}, x_4]$*. This last point indicates the non-q-integer linearity of* $p$ *over* $\mathbb{Z}[q, q^{-1}]$*, even before starting the third stage.*

Once more, similar to Remark 3.10, the above algorithm can be easily modified so as to determine the $q$-integer linearity of a given polynomial only. In other words, the algorithm can exit early and return a negative answer whenever one of the following situations occurs.

- In Step 4 or in any iteration step of Step 8.2, any of the triggers listed in Remark 3.10 is touched.
- In Step 6, the polynomial $g$ turns out to be not $q$-integer linear.

## 5. Complexity comparison

In this section, we give complexity analyses for the two algorithms presented in Sections 3 and 4 in the case of $\mathsf{R} = \mathbb{Z}[q, q^{-1}]$. In addition, we discuss two more algorithms for the same purpose, namely for computing the $q$-integer linear decomposition of polynomials, along with their costs in the bivariate case for the sake of comparison.

### *5.1. Complexity background*

We first collect some classical complexity notations and facts needed in this paper. More background on these can be found in (von zur Gathen and Gerhard, 2013).

Although our algorithms work in more general UFDs, we confine our complexity analysis to the case of integer (Laurent) polynomials, that is, when $\mathsf{D}$ is the ring of integers $\mathbb{Z}$ and then $\mathsf{R}$ is equal to $\mathbb{Z}[q, q^{-1}]$. Here $q$ can be viewed as a variable in addition to $x_1, \ldots, x_n$. Note that operations in $\mathbb{Z}[q, q^{-1}]$ can be easily transferred to those in $\mathbb{Z}[q]$ with a negligible cost. The cost is given in terms of number of word operations used so that growth of coefficients comes into play. Recall that the *word length* of a nonzero integer $a \in \mathbb{Z}$ is defined as $\mathsf{O}(\log |a|)$. In this paper, all complexity is analyzed in terms of a function $\mathsf{M}(d)$ which bounds the cost required to multiply two integers of word length at most $d$ or polynomials of degree at most $d$. We take $\mathsf{M}(d) = d^2$ using classical arithmetic and $\mathsf{M}(d) = \mathsf{O}^{\sim}(d)$ using fast arithmetic, where the *soft-Oh notation* "$\mathsf{O}^{\sim}$" is basically "O" but suppressing logarithmic factors (see (von zur Gathen and Gerhard, 2013, Definition 25.8) for a precise definition). We assume that $\mathsf{M}$ is subadditive, superlinear and subquadratic, that is, $\mathsf{M}(a) + \mathsf{M}(b) \leq \mathsf{M}(a + b)$ and $a\mathsf{M}(b) \leq \mathsf{M}(ab) \leq a^2\mathsf{M}(b)$ for all $a, b \in \mathbb{N}$.

Throughout this paper, we define the *max-norm* $\|p\|_\infty$ of a Laurent polynomial $p \in \mathbb{Z}[q, q^{-1}]$ as the maximum absolute value of its coefficients with respect to $q$, and the *max-norm* $\|p\|_\infty$ of a polynomial $p = \sum_{\boldsymbol{i}\in\mathbb{N}} p_{i_1,\ldots,i_n}\boldsymbol{x}^{\boldsymbol{i}} \in \mathbb{Z}[q, q^{-1}][\boldsymbol{x}]$ as $\max_{\boldsymbol{i}\in\mathbb{N}}\{\|p_{i_1,\ldots,i_n}\|_\infty\}$. The GCD computation is fundamental for our algorithms. Before analyzing the algorithm, let us recall some useful complexity results on GCD computation.

**Lemma 5.1** ((Gel′fond, 1960, Page 135-139))**.** *Let $p_1, \ldots, p_m \in \mathbb{Z}[\boldsymbol{x}]$. Let $p = p_1 \cdots p_m$ and let $d_i = \deg_{x_i}(p)$ for all $i = 1, \ldots, n$. Then*

$$\|p_1\|_\infty \cdots \|p_m\|_\infty \leq e^{d_1+\cdots+d_n}\|p\|_\infty,$$

*where $e$ is the base of the natural logarithm.*

Note that when $n = 1$ the above bound is actually worse than Mignotte's factor bound for large $d$, which, however, leads to the same order of magnitude for word lengths of the max-norms.

The lemma below provides bounds for the resultant of two multivariate integer polynomials, which can be verified by following the proof of (Bistritz and Lifshitz, 2010, Theorem 10) but arguing from the perspective of multivariate polynomials.

**Lemma 5.2.** *Let $f, g \in \mathbb{Z}[\boldsymbol{x}]$ with $\deg_{x_i}(f), \deg_{x_i}(g) \leq d_i$ for all $i = 1, \ldots, n$. Then*

$$\|\operatorname{Res}_{x_n}(f, g)\|_\infty \leq (2d_n)!(d_1 + 1)^{2d_n-1} \cdots (d_{n-1} + 1)^{2d_n-1}\|f\|_\infty^{d_n}\|g\|_\infty^{d_n}.$$

The next result must be known in the literature, but we could not find a convenient reference, so a proof is provided.

**Lemma 5.3.** *Let $f, g \in \mathbb{Z}[\boldsymbol{x}]$ with $\deg_{x_i}(f), \deg_{x_i}(g) \leq d_i$ for all $i = 1, \ldots, n$ and $\|f\|_\infty, \|g\|_\infty \leq \beta$. Let $d = \max\{d_1, \ldots, d_n\}$ and $D_n = d_1 \cdots d_n$. Then computing $\gcd(f, g)$ over $\mathbb{Z}$ takes $\mathrm{O}(D_n\mathsf{M}(nd + \log\beta)\log(nd + \log\beta))$ word operations.*

*Proof.* We proceed to compute $h = \gcd(f, g)$ by a small prime modular algorithm. By Lemma 5.1, $\|\gcd(f, g)\|_\infty \leq e^{d_1+\cdots+d_n}\beta \leq e^{nd}\beta = B$ with $e$ being the base of the natural logarithm. Then $\log B \in \mathrm{O}(nd + \log\beta)$. Let $k = \lceil 2\log_2((2d)!(d + 1)^{(n-1)(2d-1)}\beta^{2d})\rceil$. By Lemma 5.2, the value $k$ is an upper bound on $2\log_2\|\operatorname{Res}_{x_n}(f/h, g/h)\|_\infty$ and thus guarantees that at least $k/2$ of the first $k$ primes $p_1 = 2, \ldots, p_k$ do not divide $\operatorname{Res}_{x_n}(f/h, g/h)$. This means that at least half of the primes $p_1, \ldots, p_k$ are "lucky". It is then sufficient to choose $\lceil\log_2(2B + 1)\rceil \leq k/2$ "lucky" ones from these $k$ primes, each of word length $\mathrm{O}(\log k)$. For every chosen prime $p$, we then reduce all coefficients of $f$ and $g$ modulo $p$, using $\mathrm{O}(D_n \log\beta \log p)$ word operations, and compute $\gcd(f_p, g_p)$ with $f_p = f \mod p$ and $g_p = g \mod p$. The desired $\gcd(f, g)$ can be recovered by a final application of the Chinese remainder theorem, which takes $\mathrm{O}(D_n\mathsf{M}(nd + \log\beta)\log(nd + \log\beta))$ word operations. Neglecting the cost of computing primes, it remains to count the number of arithmetic operations, denoted by $G_p(n, d, D_n)$, used by the gcd computation in the field $\mathbb{Z}_p$ for each prime $p$, with the rest following by the fact that each operation of these takes $\mathrm{O}(M(\log p))$ word operations and $\log p \in \mathrm{O}(\log(nd) + \log\log\beta)$.

For each prime $p$, we compute $\gcd(f_p, g_p)$ with $f_p = f \mod p$ and $g_p = g \mod p$ by an evaluation-interpolation scheme (Geddes et al., 1992): evaluate coefficients of $f_p, g_p$ with respect to $x_1, \ldots, x_{n-1}$ at $d_n$ points from $\mathbb{Z}_p$ for $x_n$; compute $d_n$ GCDs over $\mathbb{Z}_p$ of two $(n - 1)$-variate polynomials of degrees at most $d_1, \dots, d_{n-1}$ in $x_1, \dots, x_{n-1}$, respectively; recover the final GCD by interpolation. Notice that there are at most $d_1 \cdots d_{n-1} = D_n/d_n$ monomials in $x_1, \dots, x_{n-1}$ appearing in each of the polynomials $f_p$ and $g_p$. The process of evaluation and interpolation then takes $\mathrm{O}((D_n/d_n)\mathsf{M}(d_n)\log d_n)$ arithmetic operations in the field $\mathbb{Z}_p$. The second step uses $\mathrm{O}(d_n G_p(n-1, d^{(n-1)}, D_{n-1}))$ arithmetic operations in $\mathbb{Z}_p$, where $d^{(n-1)} = \max\{d_1, \dots, d_{n-1}\}$ and $D_{n-1} = d_1 \cdots d_{n-1}$. Thus we obtain the recurrence relation

$$\mathrm{O}(G_p(n, d, D_n)) \subset \mathrm{O}((D_n/d_n)\mathsf{M}(d_n)\log d_n) + \mathrm{O}(d_n G_p(n-1, d^{(n-1)}, D_{n-1})).$$

From the initial condition that $G_p(1, d_1, d_1)$ is in $\mathrm{O}(\mathsf{M}(d_1)\log d_1)$, one concludes that $G_p(n, d, D_n)$ is in $\mathrm{O}((D_n/d)\mathsf{M}(d)\log D_n)$. □

*5.2. Cost analyses of our two algorithms*

We are now ready to present the cost of our first approach. In order to make it ready to use in the subsequent analysis of our second approach, we analyze the cost in the case of $\mathsf{R} = \mathbb{Z}[q, q^{-1}, z_1, \dots, z_v]$, where $v \in \mathbb{N}$ is arbitrary but fixed and the $z_i$ are additional parameters independent of $q, x_1, \dots, x_n$.

**Theorem 5.4.** *Let $p \in \mathbb{Z}[q, q^{-1}, z_1, \dots, z_v][\boldsymbol{x}]$. Assume that both the numerator and denominator of $p$ have maximum degree $d$ in each variable from $\{q, z_1, \dots, z_v, x_1, \dots, x_n\}$ separately, and let $\|p\|_\infty = \beta$. Then the algorithm* **MultivariateQILD**$_1$ *computes the $q$-integer linear decomposition of $p$ over $\mathbb{Z}[q, q^{-1}, z_1, \dots, z_v]$ using*

$$\mathrm{O}(n!d^{2n+v+2}\mathsf{M}((n^3+nv)d + n\log\beta)\log((n^2+v)d+\log\beta) + n!d^{n\lfloor n/2\rfloor}\mathsf{M}(n\log d)\log\log d)$$

*word operations.*

*Proof.* Let $T(n, d, \log\beta)$ denote the number of word operations used by the algorithm applied to the polynomial $p$. Steps 1 and 5 treat the trivial case, taking no word operations. In Step 2, finding the content $c$ amounts to computing a GCD of at most $(d+1)^n$ polynomials in $\mathbb{Z}[q, z_1, \dots, z_v]$ of degree at most $d$ in each variable separately and max-norm at most $\beta$. Thus by Lemma 5.3, this step takes $\mathrm{O}(d^{n+v+1}\mathsf{M}((v+1)d+\log\beta)\log((v+1)d+\log\beta))$ word operations. Step 3 deals with the univariate case, yielding that the initial cost $T(1, d, \log\beta)$ is in $\mathrm{O}(d^{v+2}\mathsf{M}((v+1)d+\log\beta)\log((v+1)d+\log\beta))$.

In Step 4, at each iteration of the loop, the computation of the content $g$ and its primitive part in Step 4.1 can be done using $\mathrm{O}(d^{n+v+1}\mathsf{M}((n+v)d+\log\beta)\log((n+v)d+\log\beta))$; while Step 4.2 takes $\mathrm{O}(T(n-1, d, nd+\log\beta))$ word operations as $g \in \mathbb{Z}[q, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n]$ of maximum degree at most $d$ in each variable separately and max-norm of word length $\mathrm{O}(nd + \log\beta)$ by Lemma 5.1. Since there are $n$ iterations, this step in total takes $\mathrm{O}(nd^{n+v+1}\mathsf{M}((n+v)d + \log\beta)\log((n+v)d+\log\beta)) + \mathrm{O}(nT(n-1, d, nd+\log\beta))$ word operations.

The computation of the Newton polytope of $f$ dominates the other costs in Steps 6-7, which, by (Goodman et al., 2018, Theorem 26.3.1), takes $\mathrm{O}((s\log s + s^{\lfloor n/2\rfloor})\mathsf{M}(\log d)\log\log d)$ word operations with $s$ denoting the cardinality of $\mathrm{supp}(f)$. Since $s \le (d+1)^n$, we obtain the total cost $\mathrm{O}((nd^n\log d + d^{n\lfloor n/2\rfloor})\mathsf{M}(\log d)\log\log d)$ for Steps 6-7. In Step 8, for each $\boldsymbol{\lambda} \in \Lambda$, a direct calculation shows that $f(x_1^{\lambda_n}, \dots, x_{n-1}^{\lambda_n}, yx_1^{-\lambda_1}\cdots x_{n-1}^{-\lambda_{n-1}})$ has degree in $y$ at most $d$, max-norm of word length $\mathrm{O}(nd + \log\beta)$ and at most $(d+1)^n$ nonzero monomials in $x_1, \dots, x_{n-1}$ appearing. Thus by Lemma 5.3, Step 8.1 takes $\mathrm{O}(d^{n+v+2}\mathsf{M}((n+v+2)d+\log\beta)\log((n+v+2)d+\log\beta))$ word operations, which dominates the cost for Step 8.2. Since there are at most $s - 1 \le (d+1)^n - 1$

elements in the set $\Lambda$, this step takes $\mathsf{O}(d^{2n+v+2}\mathsf{M}((n+v+2)d+\log\beta)\log((n+v+2)d+\log\beta))$ word operations. Steps 9 and 10 both take no word operations without expanding the product.

In summary, we obtain the recurrence relation

$$\begin{aligned}\mathsf{O}(T(n,d,\log\beta)) \subset \mathsf{O}(d^{2n+v+2}\mathsf{M}((n+v+2)d+\log\beta)\log((n+v+2)d+\log\beta)\\ + d^{n\lfloor n/2\rfloor}\mathsf{M}(\log d)\log\log d) + \mathsf{O}(nT(n-1,d,nd+\log\beta)),\end{aligned}$$

along with $T(1,d,\log\beta) \in \mathsf{O}(d^{v+2}\mathsf{M}((v+1)d+\log\beta)\log((v+1)d+\log\beta))$. The cost follows. $\square$

**Corollary 5.5.** *With the assumptions of Theorem 5.4, further let $v = 0$. Then the algorithm* **MultivariateQILD**$_1$ *computes the $q$-integer linear decomposition of $p$ over $\mathbb{Z}[q,q^{-1}]$ using $\mathsf{O}\tilde{~}(n!d^{2n+4}+d^{2n+2}\log^2\beta+n!d^{n\lfloor n/2\rfloor})$ word operations with classical arithmetic and $\mathsf{O}\tilde{~}(n!d^{2n+3}+n!d^{2n+2}\log\beta+n!d^{n\lfloor n/2\rfloor})$ with fast arithmetic.*

In the case of our second algorithm we have the following cost.

**Theorem 5.6.** *Let $p \in \mathbb{Z}[q,q^{-1}][\boldsymbol{x}]$. Assume that both the numerator and denominator of $p$ have maximum degree $d$ in each variable from $\{q,x_1,\ldots,x_n\}$ separately, and let $\|p\|_\infty = \beta$. Then the algorithm* **MultivariateQILD**$_2$ *computes the $q$-integer linear decomposition of $p$ over $\mathbb{Z}$ using $\mathsf{O}(d^{n+4}\mathsf{M}(n^4d+n^2\log\beta)\log(n^2d+\log\beta))$ word operations.*

*Proof.* Let $T(n,d,\log\beta)$ denote the number of word operations used by the algorithm applied to the polynomial $p$. The first three steps are exactly the same as the algorithm **MultivariateQILD**$_1$. Thus, as before, Step 1 takes no word operations, Step 2 uses $\mathsf{O}(d^{n+1}\mathsf{M}(d+\log\beta)\log(d+\log\beta))$ word operations, and Step 3 gives the initial cost $T(1,d,\log\beta) \in \mathsf{O}(d^2\mathsf{M}(d+\log\beta)\log(d+\log\beta))$. Step 4 deals with the bivariate case. By Theorem 5.4 with $n = 2$ and $v = 0$, this step yields that $T(2,d,\log\beta)$ is in $\mathsf{O}(d^6\mathsf{M}(d+\log\beta)\log(d+\log\beta))$.

In Step 5, by Lemma 5.3, the computation of the content and primitive part can be done within $\mathsf{O}(d^{n+1}\mathsf{M}(nd+\log\beta)\log(nd+\log\beta))$ word operations. Notice that $g \in \mathbb{Z}[q,x_3,\ldots,x_n]$ has maximum degree at most $d$ in each variable separately and max-norm of word length $\mathsf{O}(nd+\log\beta)$ by Lemma 5.1. Then Step 6 takes $\mathsf{O}(T(n-2,d,nd+\log\beta))$ word operations. Step 7 takes linear time in the cardinality of $\mathrm{supp}(f)$, which is at most $(d+1)^n$. In Step 8, notice that for the $k$th iteration, the polynomial $h \in \mathbb{Z}[q,x_{k+2},\ldots,x_n][y,x_{k+1}]$ has maximum degree at most $d$ in each variable separately and max-norm of word length $\mathsf{O}(nd+\log\beta)$. Thus by Theorem 5.4 with $n = 2$ and $v = n-k-1$, the $k$th iteration requires $\mathsf{O}(d^{n-k+5}\mathsf{M}((n-k-1)d+\log\beta)\log((n-k-1)d+\log\beta))$ word operations. Since $1 \le k \le n-1$, this step in total takes $\mathsf{O}(d^{n+4}\mathsf{M}(n^2d+n\log\beta)\log(nd+\log\beta))$ word operations, dominating the costs of Steps 9-10.

In summary, we obtain the recurrence relation

$$\mathsf{O}(T(n,d,\log\beta)) \subset \mathsf{O}(d^{n+4}\mathsf{M}(n^2d+n\log\beta)\log(nd+\log\beta)) + \mathsf{O}(T(n-2,d,nd+\log\beta)),$$

along with $T(1,d,\log\beta) \in \mathsf{O}(d^2\mathsf{M}(d+\log\beta)\log(d+\log\beta))$ and $T(2,d,\log\beta) \in \mathsf{O}(d^6\mathsf{M}(d+\log\beta)\log(d+\log\beta))$. The announced cost follows. $\square$

**Corollary 5.7.** *With the assumptions of Theorem 5.6, the algorithm* **MultivariateQILD**$_2$ *then computes the $q$-integer linear decomposition of $p$ over $\mathbb{Z}[q,q^{-1}]$ using $\mathsf{O}(d^{n+6}+d^{n+4}\log^2\beta)$ word operations with classical arithmetic and $\mathsf{O}\tilde{~}(d^{n+5}+d^{n+4}\log\beta)$ with fast arithmetic.*

**Remark 5.8.** *The complexity of our second approach could be further improved if one find a multivariate version of the GCD algorithm of Conflitti (2003). This is the algorithm which randomly reduces computing the GCD of several polynomials over a finite field to computing a single GCD of two polynomials over the same field.*

*5.3. Cost analysis of the resultant-based algorithm*

In this subsection, we review the algorithm of Le (2001), which is based on resultants and serves as a $q$-analogue of the algorithm of Abramov and Le (2002) in the ordinary shift case. As already mentioned in the introduction, this algorithm is completely focused on bivariate polynomials. So we will further extend it to also tackle polynomials having more than two variables.

As we proceed with our first approach, the algorithm of Le (2001) first finds candidates for $q$-integer linear types of a given bivariate polynomial and then obtains the corresponding univariate polynomials by going through these candidates. The difference is that it uses resultants to determine candidates and performs bivariate GCD computations for detecting each candidate.

In order to state its main idea, let $p \in \mathsf{R}[x, y]$ be a polynomial of positive total degree which is primitive with respect to its either variable. By Lemma 4.2, an integer pair $(\lambda, \mu)$ with $\lambda\mu \neq 0$ is a $q$-integer linear type of $p$ if and only if there exists a factor $f \in \mathsf{R}[x, y] \setminus \mathsf{R}$ of $p$ with the property that $f$ divides $f(q^{\mu}x, q^{-\lambda}y)$ in $\mathsf{R}[x, y]$. Note that such an $f$ must satisfy $\deg_x(f)\deg_y(f) > 0$ and $f(x, 0)f(0, y) \neq 0$ because $p$ is assumed to be primitive with respect to its either variable. By a careful study on the structure of the factor $f$, it is then not hard to see that $f$ divides $f(q^{\mu}x, q^{-\lambda}y)$ in $\mathsf{R}[x, y]$ if and only if $f$ divides $f(qx, q^{-\lambda/\mu}y)$ in $\mathsf{R}[x, y]$. Observe that any integer pair $(\lambda, \mu)$ with $\lambda\mu \neq 0$ is uniquely determined by the rational $r = -\lambda/\mu$. We have thus shown the following.

**Lemma 5.9.** *With $p$ given above, a nonzero rational number $r$ gives rise to a $q$-integer linear type of $p$ if and only if* $\gcd(p, p(qx, q^r y)) \notin \mathsf{R}$.

This implies that for any integer-linear type $(\lambda, \mu)$ of $p$ with $\lambda\mu \neq 0$, the rational number $-\lambda/\mu$ must be a root of the resultant $\mathrm{Res}_y(p, p(qx, q^r y)) \in \mathsf{R}[q^r, x]$ in terms of $r$, or equivalently, it is eliminated by the content in $\mathsf{R}[q^r]$ of the resultant with respect to $x$. Note that such a rational root of a polynomial in $\mathsf{R}[q^r]$ can be found by matching powers of $q$ appearing in the given polynomial in pairs along with a subsequent substitution for zero testing. One can find more details in (Le, 2001, §5). Accordingly, we derive a way to produce candidates for the rationals $-\lambda/\mu$ (and then the $q$-integer linear types $(\lambda, \mu)$). After generating candidates, the algorithm of Le (2001) continues to compute the possible corresponding univariate polynomial for each candidate $r = -\lambda/\mu$ by finding a factor $f$ of $p$ that stabilizes $\gcd(f, f(qx, q^r y))$, or more efficiently, $\gcd(f, f(q^{\mu}x, q^{-\lambda}y))$. This operation actually induces bivariate polynomial arithmetic over $\mathsf{R}$ and thus may take considerably more time than Step 8.1 of our algorithm **MultivariateQILD**$_1$. In order to improve the performance, we instead proceed by using Step 8 of our algorithm.

We remark that Lemma 5.9 cannot be literally carried over to polynomials in more than two variables. It is actually not clear how to directly generalize the algorithm of Le (2001) to the multivariate case. Nevertheless, using the bivariate-based scheme indicated by Proposition 4.3, this algorithm extends to the case of polynomials in any number of variables in the same fashion as our second approach.

The following theorem gives a complexity analysis for the algorithm of Le (2001) when applied to a polynomial in $\mathbb{Z}[q, q^{-1}][x, y]$.

**Theorem 5.10.** *Let $p \in \mathbb{Z}[q, q^{-1}][x, y]$. Assume that both the numerator and denominator of $p$ have maximum degree $d$ in each variable from $\{q, x, y\}$ separately, and let $\|p\|_{\infty} = \beta$. Then the algorithm of Le takes* $\mathsf{O}((d^6 \log d + d^6 \log\beta)\mathsf{M}(d^2)\mathsf{M}(\log d + \log\log\beta) \log d \log(\log d + \log\log\beta) + d^6\mathsf{M}(d \log d + d \log\beta) \log(d \log d + d \log\beta))$ *word operations.*

*Proof.* With a slight abuse of notation, let $p$ be the input polynomial with content with respect to its either variable being removed. Then $p \in \mathbb{Z}[q, x, y]$ and $\log \|p\|_{\infty} \in \mathsf{O}(d + \log\beta)$. The algorithm proceeds to compute the resultant $\mathrm{Res}_y(p, p(qx, q^r y))$ with $r$ undetermined. By definition, it is readily seen that $\mathrm{Res}_y(p, p(qx, q^r y))$ is a polynomial in $\mathbb{Z}[q, q^r, x]$ of degree in $q$ at most $3d^2$, degree in $q^r$ at most $d^2$ and degree in $x$ at most $2d^2$. Observe that every entry in the Sylvester matrix is a monomial in $q^r$. Thus we have $\| \mathrm{Res}_y(p, p(qx, q^r y))\|_\infty \le \| \mathrm{Res}_y(p, p(qx, y))\|_\infty$, which, by Lemma 5.2, is at most $B = (2d)!(2d + 1)^{2d-1}(d + 1)^{2d-1}\|p\|_\infty^{2d}$. Then $\log B \in \mathrm{O}(d \log d + d \log \beta)$. Viewing $q^r$ as a new indeterminate $u$ independent of $q$, we can compute this resultant using a small prime modular algorithm, along with an evaluation-interpolation scheme: (1) choose $\lceil \log_2(2B + 1)\rceil$ primes, each of word length $\mathrm{O}(\log \log B)$; (2) for every chosen prime $h$, do the following: reduce all coefficients of $p$ and $p(qx, uy)$ modulo $h$, evaluate both modular images successively at $3d^2$ points for $q$, $d^2$ points for $u$ and $d^2$ points for $x$, compute $6d^6$ resultants of two polynomials in $\mathbb{Z}_h[y]$ of degrees in $y$ at most $d$, and recover the modular resultant by interpolation; (3) reconstruct the desired resultant using the Chinese remainder theorem. Neglecting the cost for choosing primes in Step (1), we analyze the costs used by Steps (2)-(3). In Step (2), the cost per prime $h$ for reducing all coefficients modulo $h$ is $\mathrm{O}(d^2 \log \beta \log h)$ word operations. The process of evaluation and interpolation is performed in $\mathrm{O}(d^5 \mathsf{M}(d^2) \log d)$ arithmetic operations in $\mathbb{Z}_h$. Each resultant over $\mathbb{Z}_h[y]$ can be computed using $\mathrm{O}(\mathsf{M}(d) \log d)$ arithmetic operations in $\mathbb{Z}_h$, yielding $\mathrm{O}(d^6 \mathsf{M}(d) \log d)$ arithmetic operations in $\mathbb{Z}_h$ in total for this step. Notice that the cost for each arithmetic operation in $\mathbb{Z}_h$ is $\mathrm{O}(\mathsf{M}(\log h) \log \log h)$ word operations. Also notice that every chosen prime $h$ is of word length $\log h \in \mathrm{O}(\log d + \log \log \beta)$. Thus Step (2) in total takes $\mathrm{O}((d^6 \log d + d^6 \log \beta)\mathsf{M}(d^2)\mathsf{M}(\log d + \log \log \beta) \log d \log(\log d + \log \log \beta))$ word operations. In Step (3), the Chinese remainder theorem requires $\mathrm{O}(d^6 \mathsf{M}(d \log d + d \log \beta) \log(d \log d + d \log \beta))$ word operations. Therefore, computing the resultant $\mathrm{Res}_y(p, p(qx, q^r y))$ takes $\mathrm{O}((d^6 \log d + d^6 \log \beta)\mathsf{M}(d^2)\mathsf{M}(\log d + \log \log \beta) \log d \log(\log d + \log \log \beta) + d^6 \mathsf{M}(d \log d + d \log \beta) \log(d \log d + d \log \beta))$ word operations. This dominates the costs for subsequent steps including finding the rational roots and computing corresponding univariate polynomials. The claimed cost follows. □

**Corollary 5.11.** *With the assumptions of Theorem 5.10, the algorithm of Le takes* $\mathrm{O}^\sim(d^{10} \log \beta + d^8 \log^2 \beta)$ *word operations with classical arithmetic and* $\mathrm{O}^\sim(d^8 \log \beta)$ *with fast arithmetic.*

*5.4. Cost analysis of the factorization-based algorithm*

In the current subsection, we introduce another algorithm which is based on full irreducible factorization of polynomials and works for polynomials in any number of variables. This algorithm can be viewed as a $q$-analogue of the algorithm of Li and Zhang (2013) from the ordinary shift case. In order to analyze its cost, we will briefly describe its main ideas.

The key observation is that, for any $q$-integer linear polynomial $p \in \mathsf{R}[\boldsymbol{x}]$ of only one type $(\lambda_1, \ldots, \lambda_n)$, the difference of any two vectors from $\mathrm{supp}(p)$ can be written into the form $k \cdot (\lambda_1, \ldots, \lambda_n)$ for some $k \in \mathbb{Z}$. This allows one to readily determine the $q$-integer linearity of any irreducible polynomial. That is, given an irreducible polynomial $p \in \mathsf{R}[\boldsymbol{x}]$, take $\boldsymbol{\alpha} \in \mathbb{N}^n$ to be the minimal vector of $\mathrm{supp}(p)$ and investigate whether the difference between $\boldsymbol{\alpha}$ and any other vector from $\mathrm{supp}(p)$ is equal to a scalar multiple of the same integer vector. One thus immediately establishes a factorization-based algorithm for computing the $q$-integer linear decomposition of a polynomial in $\mathsf{R}[\boldsymbol{x}]$: (1) first perform the full irreducible factorization of the input polynomial over $\mathsf{R}$, then (2) determine the $q$-integer linearity of each irreducible factor and finally (3) regroup all factors of the same $q$-integer linear type.

A careful study of the above algorithm leads to the following complexity.

**Theorem 5.12.** *Let $p$ be a polynomial in $\mathbb{Z}[q, q^{-1}][x, y]$. Assume that both the numerator and denominator of $p$ have maximum degree $d$ in each variable from $\{q, x, y\}$ separately, and let*

$\|p\|_\infty = \beta$. *Then the factorization-based algorithm described above requires* $\mathrm{O}^\sim(d^9 \log^2 \beta)$ *word operations with classical arithmetic and* $\mathrm{O}^\sim(d^8 \log \beta)$ *with fast arithmetic.*

*Proof.* Computing a complete factorization of $p$ into irreducibles over $\mathbb{Z}[q, q^{-1}]$ dominates the other costs of the algorithm. This is essentially the complexity of factoring in $\mathbb{Z}[q][x, y]$, for polynomials bounded by degree $d$ in all variables ($q$, $x$ and $y$). While we do not know of an explicit analysis of this complexity (beyond being in polynomial-time, since (Kaltofen, 1985)), the algorithm of Gao (2003) can be applied and analyzed over the function field $\mathbb{Q}(q)$, and appears to require $\mathrm{O}^\sim(d^9 \log^2 \beta)$ word operations with classical arithmetic and $\mathrm{O}^\sim(d^8 \log \beta)$ with fast arithmetic. □

**Remark 5.13.** *Recall from Corollary 5.11 that the algorithm of Le takes* $\mathrm{O}^\sim(d^{10} \log \beta + d^8 \log^2 \beta)$ *word operations with classical arithmetic and* $\mathrm{O}^\sim(d^8 \log \beta)$ *with fast arithmetic. This compares to the factorization-based algorithm described above which requires* $\mathrm{O}^\sim(d^9 \log^2 \beta)$ *word operations with classical arithmetic and* $\mathrm{O}^\sim(d^8 \log \beta)$ *with fast arithmetic. All of these compare to Corollary 5.5 (or Corollary 5.7) with* $n = 2$*, which reads that our algorithms when restricted to the bivariate case take* $\mathrm{O}^\sim(d^8 + d^6 \log^2 \beta)$ *word operations with classical arithmetic and* $\mathrm{O}^\sim(d^7 + d^6 \log \beta)$ *with fast arithmetic.*

## 6. Implementation and timings

We have implemented both of our algorithms in Maple 2018 in the case where the domain $\mathsf{R}$ is the ring of polynomials over $\mathbb{Z}[q, q^{-1}]$. The code is available by email request. In order to get an idea about the efficiency of our algorithms, we have compared their runtimes, as well as the memory requirements, to the performance of our Maple implementations of the two algorithms discussed in the preceding section.

The test suite was generated by

$$p = P_0 \prod_{i=1}^{m} \mathrm{num}(P_i(\boldsymbol{x}^{\boldsymbol{\lambda}_i})), \tag{6.1}$$

where $n, m \in \mathbb{N}$,

- $P_0 \in \mathbb{Z}[q][x_1, \ldots, x_n]$ is a random polynomial with $\deg_{x_1,\ldots,x_n}(P_0) = \deg_q(P_0) = d_0$,
- the $\boldsymbol{\lambda}_i \in \mathbb{Z}^n$ are random integer vectors each of which has entries of maximum absolute value no more than 10 (note that they may not be distinct),
- $P_i(z) = f_{i1}(z) f_{i2}(z)$ with $f_{ij}(z) \in \mathbb{Z}[q][z]$ a random polynomial of degree $j \cdot d$ for some $d \in \mathbb{N}$, and $\mathrm{num}(\cdots)$ denotes the numerator of the argument.

Note that, in all tests, the algorithms take the expanded forms of examples given above as input. All timings are measured in seconds on a Linux computer with 128GB RAM and fifteen 1.2GHz Dual core processors. The computations for the experiments did not use any parallelism.

For a selection of random polynomials of the form (6.1) for different choices of $n, m, d_0, d$, Table 1 collects the timings of the algorithm of Le (LQILD), the algorithm based on factorization (FQILD) and our two algorithms ($\mathrm{MQILD}_1$, $\mathrm{MQILD}_2$). The dash in the table indicates that with this choice of $(m, n, d_0, d)$, the corresponding procedure reached the CPU time limit (which was set to 12 hours) and yet did not return.

| $(n,m,d_0,d)$ | LQILD | FQILD | MQILD$_1$ | MQILD$_2$ |
|---|---|---|---|---|
| $(2,1,1,1)$ | 5408.48 | 0.04 | 0.01 | 0.01 |
| $(2,1,5,1)$ | 8381.99 | 0.06 | 0.03 | 0.03 |
| $(2,1,10,1)$ | – | 0.19 | 0.04 | 0.04 |
| $(2,1,20,1)$ | – | 0.63 | 0.09 | 0.09 |
| $(2,1,30,1)$ | – | 1.47 | 0.13 | 0.10 |
| $(2,1,40,1)$ | – | 2.55 | 0.24 | 0.21 |
| $(2,1,50,1)$ | – | 6.64 | 0.42 | 0.39 |
| $(2,2,10,1)$ | – | 0.92 | 0.10 | 0.08 |
| $(2,3,10,1)$ | – | 3.29 | 0.31 | 0.26 |
| $(2,4,10,1)$ | – | 5.74 | 0.67 | 0.54 |
| $(2,5,10,1)$ | – | 18.83 | 2.01 | 1.54 |
| $(2,2,10,2)$ | – | 4.55 | 0.27 | 0.20 |
| $(2,4,10,2)$ | – | 114.82 | 4.98 | 4.53 |
| $(2,5,10,2)$ | – | 264.02 | 25.63 | 24.29 |
| $(2,3,10,2)$ | – | 36.14 | 1.38 | 1.21 |
| $(2,3,10,3)$ | – | 169.13 | 4.28 | 3.80 |
| $(2,3,10,4)$ | – | 649.03 | 12.15 | 12.86 |
| $(2,3,10,5)$ | – | 1554.31 | 31.54 | 33.50 |
| $(2,2,5,1)$ | – | 0.32 | 0.05 | 0.05 |
| $(3,2,5,1)$ | – | 1.99 | 0.14 | 0.12 |
| $(4,2,5,1)$ | – | 11.46 | 0.35 | 0.20 |
| $(5,2,5,1)$ | – | 183.17 | 0.99 | 0.63 |
| $(6,2,5,1)$ | – | 1141.32 | 2.58 | 0.98 |
| $(7,2,5,1)$ | – | 11759.89 | 6.07 | 1.74 |
| $(8,2,5,1)$ | – | 18153.45 | 10.60 | 5.29 |
| $(9,2,5,1)$ | – | – | 65.53 | 38.12 |
| $(10,2,5,1)$ | – | – | 176.25 | 89.87 |

Table 1: Comparison of all four algorithms for a collection of polynomials $p$ of the form (6.1).

## 7. Conclusion

In this paper we have presented two new algorithms for computing the $q$-integer linear decomposition of a multivariate polynomial over any UFD of characteristic zero. When restricted to the bivariate case, both algorithms reduce to the same algorithm. For the sake of comparison, we included an algorithm based on full irreducible factorization of polynomials. Compared with the known algorithm of Le (2001) and this factorization-based algorithm in the bivariate case, our algorithm is considerably faster. In practice, both our algorithms are also more efficient than these two algorithms. In addition, we have extended and improved the original contribution of Le and provided complexity analysis for the improved version. We remark that both our algorithms have much better performances than the other two algorithms in the case where the coefficient domain contains algebraic numbers.

## Acknowledgments

We would like to thank the anonymous referees for many useful and constructive suggestions, which considerably improved the structure of this work. Most of the work presented in this paper was carried out while Hui Huang was a Post Doctoral Fellow at the University of Waterloo. This research was partly supported by the Natural Sciences and Engineering Research Council (NSERC) Canada. Hui Huang was also supported by the Fundamental Research Funds for the Central Universities.

## References

Abramov, S. A., Le, H. Q., 2002. A criterion for the applicability of Zeilberger's algorithm to rational functions. Discrete Math. 259 (1-3), 1–17.
URL http://dx.doi.org/10.1016/S0012-365X(02)00442-9

Abramov, S. A., Petkovšek, M., 2002. On the structure of multivariate hypergeometric terms. Adv. in Appl. Math. 29 (3), 386–411.
URL http://dx.doi.org/10.1016/S0196-8858(02)00022-2

Bistritz, Y., Lifshitz, A., 2010. Bounds for resultants of univariate and bivariate polynomials. Linear Algebra Appl. 432 (8), 1995–2005.
URL https://doi.org/10.1016/j.laa.2009.08.012

Chen, S., Koutschan, C., 2019. Proof of the Wilf-Zeilberger conjecture for mixed hypergeometric terms. J. Symbolic Comput. 93, 133–147.
URL http://www.sciencedirect.com/science/article/pii/S0747717118300671

Chen, W. Y. C., Hou, Q.-H., Mu, Y.-P., 2005. Applicability of the $q$-analogue of Zeilberger's algorithm. J. Symbolic Comput. 39 (2), 155–170.
URL http://dx.doi.org/10.1016/j.jsc.2004.09.002

Conflitti, A., 2003. On computation of the greatest common divisor of several polynomials over a finite field. Finite Fields Appl. 9 (4), 423–431.
URL https://doi.org/10.1016/S1071-5797(03)00022-4

Du, H., Li, Z., 2019. The Ore-Sato theorem and shift exponents in the $q$-difference case. J. Syst. Sci. Complex. 32 (1), 271–286.
URL https://doi.org/10.1007/s11424-019-8355-1

Gao, S., 2003. Factoring multivariate polynomials via partial differential equations. Math. Comp. 72 (242), 801–822.
URL https://doi.org/10.1090/S0025-5718-02-01428-X

von zur Gathen, J., Gerhard, J., 2013. Modern Computer Algebra, 3rd Edition. Cambridge University Press, Cambridge.
URL http://dx.doi.org/10.1017/CBO9781139856065

Geddes, K. O., Czapor, S. R., Labahn, G., 1992. Algorithms for computer algebra. Kluwer Academic Publishers, Boston, MA.
URL http://dx.doi.org/10.1007/b102438

Gel′fond, A. O., 1960. Transcendental and algebraic numbers. Translated from the first Russian edition by Leo F. Boron. Dover Publications, Inc., New York.

Giesbrecht, M., Huang, H., Labahn, G., Zima, E., 2019. Efficient integer-linear decomposition of multivariate polynomials. In: Proceedings of ISSAC'19. ACM, New York, pp. 171–178.
URL http://doi.acm.org/10.1145/3326229.3326261

Giesbrecht, M., Huang, H., Labahn, G., Zima, E., 2021. Efficient rational creative telescoping. To appear in Journal of Symbolic Computation.

Goodman, J. E., O'Rourke, J., Tóth, C. D. (Eds.), 2018. Handbook of Discrete and Computational Geometry. Discrete Mathematics and its Applications (Boca Raton). CRC Press, Boca Raton, FL, third edition of [ MR1730156].

Grünbaum, B., 2003. Convex polytopes, 2nd Edition. Vol. 221 of Graduate Texts in Mathematics. Springer-Verlag, New York, prepared and with a preface by Volker Kaibel, Victor Klee and Günter M. Ziegler.
URL https://doi.org/10.1007/978-1-4613-0019-9

Kaltofen, E., 1985. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. SIAM J. Comput. 14 (2), 469–489.
URL https://doi.org/10.1137/0214035

Le, H. Q., 2001. On the $q$-analogue of Zeilberger's algorithm to rational functions. Program. Comput. Softw. 27 (1), 35–42.
URL http://dx.doi.org/10.1023/A:1007186703355

Le, H. Q., Abramov, S. A., Geddes, K. O., 2001. A direct algorithm to construct the minimal telescopers for rational functions (q-difference case). Technical Report CS-2001-25, Department of Computer Science, University of Waterloo, ON, Canada.

Li, Z., Zhang, Y., 2013. An algorithm for decomposing multivariate hypergeometric terms. A contributed talk in CM'13.

Ostrowski, A. M., 1921. Über die Bedeutung der Theorie der konvexen Polyeder für die formale Algebra. Jahresberichte Deutsche Math. Verein 30, 98–99.

Ostrowski, A. M., 1975. On multiplication and factorization of polynomials. I. Lexicographic orderings and extreme aggregates of terms. Aequationes Math. 13 (3), 201–228.
URL https://doi.org/10.1007/BF01836524

Paule, P., Riese, A., 1997. A Mathematica $q$-analogue of Zeilberger's algorithm based on an algebraically motivated approach to $q$-hypergeometric telescoping. In: Special functions, $q$-series and related topics (Toronto, ON, 1995). Vol. 14 of Fields Inst. Commun. Amer. Math. Soc., Providence, RI, pp. 179–210.

Wilf, H. S., Zeilberger, D., 1992. An algorithmic proof theory for hypergeometric (ordinary and "$q$") multisum/integral identities. Invent. Math. 108 (3), 575–633.
URL http://dx.doi.org/10.1007/BF02100618

Ziegler, G. M., 1995. Lectures on polytopes. Vol. 152 of Graduate Texts in Mathematics. Springer-Verlag, New York.
URL https://doi.org/10.1007/978-1-4613-8431-1