

Hard to Detect Factors of Univariate Integer Polynomials

Alberto Dennunzio ^{1,*}, Enrico Formenti ^{2,*} and Luciano Margara ^{3,*}¹ Department of Informatics, Systems and Communication, University of Milano-Bicocca, 20100 Milan, Italy² CNRS, I3S, Université Côte d'Azur, France³ Department of Computer Science and Engineering, Via dell'Università 50, 47521 Cesena, Italy

* Correspondence: alberto.dennunzio@unimib.it (A.D.); enrico.formenti@univ-cotedazur.fr (E.F.); luciano.margara@unibo.it (L.M.)

† These authors contributed equally to this work.

Abstract: We investigate the computational complexity of deciding whether a given univariate integer polynomial $p(x)$ has a factor $q(x)$ satisfying specific additional constraints. When the only constraint imposed on $q(x)$ is to have a degree smaller than the degree of $p(x)$ and greater than zero, the problem is equivalent to testing the irreducibility of $p(x)$ and then it is solvable in polynomial time. We prove that deciding whether a given monic univariate integer polynomial has factors satisfying additional properties is NP-complete in the strong sense. In particular, given any constant value $k \in \mathbb{Z}$, we prove that it is NP-complete in the strong sense to detect the existence of a factor that returns a prescribed value when evaluated at $x = k$ (Theorem 1) or to detect the existence of a pair of factors—whose product is equal to the original polynomial—that return the same value when evaluated at $x = k$ (Theorem 2). The list of all the properties we have investigated in this paper is reported at the end of Section Introduction.

Keywords: computational complexity; polynomials; factorization; NP-completeness; semirings**MSC:** 68R01; 68Q17; 13P05; 11C08

Citation: Dennunzio, A.; Formenti, E.; Margara, L. Hard to Detect Factors of Univariate Integer Polynomials. *Mathematics* **2023**, *11*, 3602. <https://doi.org/10.3390/math11163602>

Academic Editor: Alexander Grigoriev

Received: 10 July 2023

Revised: 17 August 2023

Accepted: 17 August 2023

Published: 20 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The idea of decomposing a polynomial into the product of smaller ones is definitely not new. A large amount of literature has been devoted to the factorization of polynomials (without claim of exhaustiveness, see [1–3]) as well as to the decomposition of other mathematical objects, e.g., numbers, matrices, graphs and so on. The basic idea behind factorization is decomposing a complex object into smaller and easier to analyze pieces. Properties satisfied by each piece might shed some light on the properties satisfied by the entire object. As an example, from irreducible factors of a polynomial, we can recover valuable information about its roots. As far as polynomials are concerned, much attention has been dedicated to the problem of factoring them into irreducible polynomials, i.e., into no further factorable elements. The first polynomial factorization algorithm was published by Theodor Von Schubert in 1793 [4]. Since then, dozens of papers on the computational complexity of polynomial factorization have been published. In 1982, Arjen K. Lenstra, Hendric W. Lenstra, and Laszlo Lovasz [2] published the first polynomial time algorithm for factoring polynomials over \mathbb{Q} and then over \mathbb{Z} .

When dealing with the computational complexity of problems whose input is a polynomial, it is crucial to specify the way we represent it. The standard way of representing a polynomial $p(x) = c_0 + c_1x + \dots + c_nx^n$ is by giving the list $\langle c_0, c_1, \dots, c_n \rangle$ of its coefficients. In this case, the size of the polynomial is proportional to n and does not depend on the number of zero coefficients in $p(x)$. The other way to represent a polynomial (called lacunary or sparse representation) consists of the list of nonzero monomials. Lacunary representation may lead to an exponentially shorter representation of the same polynomial

with respect to the standard notation. The computational cost of an algorithm can be polynomially bounded in the standard input size and, at the same time, exponentially large in the lacunary input size. Testing the irreducibility of lacunary polynomials or computing the greatest common divisor of two lacunary polynomials are NP-hard problems [5–7]. Computing the irreducible factors of bounded degree of lacunary polynomials can be accomplished in polynomial time [8] as well as computing the integer roots of lacunary integer polynomials [9].

In this paper, we will use standard notation for polynomials. This makes our NP-completeness results even stronger.

Irreducible factorization of polynomials, under some conditions, is unique and any other factorization into not necessarily irreducible elements can be obtained by properly grouping suitable irreducible factors. In this paper we focus our attention on some particular types of factors, not necessarily irreducible, and on the computational complexity of detecting their existence. We will prove that some types of factors are hard to detect while some others are not. In other words, we show that computing irreducible factors of a polynomial can be much easier than computing other types of factors. It turns out, as expected, that the boundary between polynomially computable factors and “hard to compute” ones is far from being completely understood.

We wish to emphasize that the main aim of this paper is not to provide technically difficult proofs of long standing open problems, but rather to show a different perspective in dealing with polynomial decomposition problems.

We face the following general problem. Given an integer polynomial $p(x) \in \mathbb{Z}[x]$ and some specific property P , decide whether $p(x)$ admits one or more factors that satisfy P .

Here is a list of problems we have analyzed in this paper.

- Q1. Let $k \in \mathbb{Z}$ be any fixed integer.
Given $p(x) \in \mathbb{Z}[x]$ and $h \in \mathbb{Z}$, decide whether there exists a factor $q(x) \in \mathbb{Z}[x]$ of $p(x)$ such that $q(k) = h$. (Theorem 1).
- Q2. Given $p(x) \in \mathbb{Z}[x]$ and $h \in \mathbb{Z}$, decide whether there exists a factor $q(x) \in \mathbb{Z}[x]$ of $p(x)$ such that the sum of all the coefficients of $q(x)$ is equal to h . (Corollary 1).
- Q3. Given $p(x) \in \mathbb{Z}[x]$ and $h \in \mathbb{Z}$, decide whether there exists a factor $q(x) \in \mathbb{Z}[x]$ of $p(x)$ such that the constant term of $q(x)$ is equal to h . (Corollary 2).
- Q4. Let $k \in \mathbb{Z}$ be any fixed integer.
Given $p(x) \in \mathbb{Z}[x]$, decide whether there exists two factors $q(x), r(x) \in \mathbb{Z}[x]$ of $p(x)$ such that $p(x) = q(x) \cdot r(x)$ and $q(k) = r(k)$. (Theorem 2).
- Q5. Given $p(x) \in \mathbb{Z}[x]$, decide whether there exists two factors $q(x), r(x) \in \mathbb{Z}[x]$ of $p(x)$ such that $p(x) = q(x) \cdot r(x)$ and the sum of all the coefficients of $q(x)$ is equal to the sum of all the coefficients of $r(x)$. (Corollary 4).
- Q6. Given $p(x) \in \mathbb{Z}[x]$, decide whether there exists two factors $q(x), r(x) \in \mathbb{Z}[x]$ of $p(x)$ such that $p(x) = q(x) \cdot r(x)$ and constant term of $q(x)$ is equal to the constant term of $r(x)$. (Corollary 3).
- Q7. Given $p(x) \in \mathbb{Z}[x]$, decide whether there exists two factors $q(x), r(x) \in \mathbb{N}[x]$ of $p(x)$ such that $p(x) = q(x) \cdot r(x)$. (Question 1).
- Q8. Given $p(x) \in \mathbb{Z}[x]$ and $h, m \in \mathbb{Z}$, decide whether there exists a factor $q(x) \in \mathbb{Z}[x]$ of $p(x)$ such that the coefficient of the monomial with degree m in $q(x)$ is equal h . (Question 2).

The rest of the paper is organized as follows. In Section 2, we give some basic definitions and known results. In Section 3, we prove that problems from Q1 to Q6 are NP-complete in the strong sense. In Section 4, we introduce and discuss open questions Q7 and Q8. Section 5 contains conclusions.

2. Definitions and Known Results

Let \mathbb{Z} denote the set of integer numbers and $\mathbb{Z}[x]$ the set of integer polynomials (polynomials with coefficients in \mathbb{Z}). Given two integer polynomials $p(x)$ and $q(x)$, we say that $q(x)$ divides $p(x)$ (we write $q(x) | p(x)$) if and only if there exists an integer polynomial

$r(x)$ such that $p(x) = q(x) \cdot r(x)$. The degree of a polynomial $p(x) = c_0 + c_1x + \dots + c_nx^n$ with $c_n \neq 0$ (denoted by $\text{deg}(p(x))$) is n . Given an integer polynomial $p(x)$, we say that $q(x) \in \mathbb{Z}[x]$ is a factor of $p(x)$ if and only if $q(x)|p(x)$. An integer polynomial with degree n is reducible if and only if it admits a factor $q(x)$ such that $0 < \text{deg}(q(x)) < n$. It is irreducible otherwise. In the rest of the paper, we will only consider monic integer polynomials, i.e., integer polynomials whose leading coefficient (coefficient of the highest degree monomial) is equal to 1.

We now introduce some well-known NP-complete computational problems that we will use for our reductions.

Definition 1 (SUBSET-SUM). Given $n + 1$ positive integers $\langle a_1, \dots, a_n, t \rangle$ decide whether there exist $I \subseteq \{1, \dots, n\}$ such that

$$\sum_{i \in I} a_i = t$$

Definition 2 (SUBSET-PRODUCT). Given $n + 1$ positive integers $\langle a_1, \dots, a_n, t \rangle$ decide whether there exist $I \subseteq \{1, \dots, n\}$ such that

$$\prod_{i \in I} a_i = t$$

Definition 3 (PRODUCT-PARTITION). Given n positive integers $\langle a_1, \dots, a_n \rangle$ decide whether there exists a partition of the set $\{1, \dots, n\}$ into two nonempty subsets I and J such that

$$\prod_{i \in I} a_i = \prod_{j \in J} a_j$$

The SUBSET-SUM problem (problem [SP13], page 224 in [10]) has been proved to be NP-complete in [11]. It is solvable in pseudo-polynomial time. The SUBSET-PRODUCT problem (problem [SP14], page 225 in [10]) has been proved to be NP-complete in the strong sense in [10,12]. The PRODUCT-PARTITION problem has been proved to be NP-complete in the strong sense in [13]. A problem is said to be NP-complete in the strong sense if it remains NP-complete even when all of its numerical parameters are bounded by a polynomial in the length of the input (see [14] for details).

3. Hard to Detect Factors

In this section, we prove that problems from Q1 to Q6 are NP-complete in the strong sense. The following observation completely characterizes any factor of a monic univariate integer polynomial with integer roots.

Remark 1. Let $n \geq 1$ and a_1, \dots, a_n be n integers. Let $p(x) \in \mathbb{Z}[x]$ be the following monic univariate integer polynomial

$$p(x) = \prod_{i=1}^n (x - a_i).$$

An integer polynomial $q(x) \in \mathbb{Z}[x]$, $q(x) \neq 1$, is a factor of $p(x)$ if and only if

$$\exists I \subseteq \{1, \dots, n\} : q(x) = \prod_{i \in I} (x - a_i)$$

Remark 2. Computing all the coefficients of an integer polynomial

$$p(x) = \prod_{i=1}^n (x - a_i)$$

takes $O(n^2)$ operations.

Proof. Let $k \in \{1, \dots, n - 1\}$. Let $p_k(x) = \prod_{i=1}^k (x - a_i)$. It is easy to verify that the degree of $p_k(x)$ is equal to k and then the number of coefficients of $p_k(x)$ is at most $k + 1$. Computing $p_{k+1}(x)$ from $p_k(x)$ takes $O(k)$ operations. Then, computing $p_n(x) = p(x)$ takes $O(n^2)$ operations. \square

Definition 4 (*k*-FACTOR problem). Let $k \in \mathbb{Z}$ be any fixed integer. The *k*-FACTOR problem is defined as follows.

Input: $p(x) \in \mathbb{Z}[x]$ and $h \in \mathbb{Z}$

Output: - YES if $p(x)$ has a factor $q(x) \in \mathbb{Z}[x]$ such that $q(k) = h$,
 - NO otherwise.

Theorem 1. For any fixed $k \in \mathbb{Z}$, *k*-FACTOR problem is NP-complete in the strong sense.

Proof. Let $k \in \mathbb{Z}$ be any fixed integer. We reduce the SUBSET-PRODUCT problem (Definition 2) to the *k*-FACTOR problem.

Let $\langle a_1, \dots, a_n, t \rangle$ be any instance of SUBSET-PRODUCT. Let $b_i = a_i - k$ for $i = 1, \dots, n$ and

$$p(x) = \prod_{i=1}^n (x + b_i)$$

Let $\langle p(x), t \rangle$ be the corresponding instance of *k*-FACTOR.

$\langle p(x), t \rangle$ is a YES instance of *k*-FACTOR if and only if there exists a factor $q(x) \in \mathbb{Z}[x]$ of $p(x)$ such that $q(k) = t$. Or equivalently, by Observation 1, if and only if

$$\exists I \subseteq \{1, \dots, n\} : q(x) = \prod_{i \in I} (x + b_i) \text{ and } q(k) = t \tag{1}$$

Equation (1) is true if and only if

$$\exists I \subseteq \{1, \dots, n\} : \prod_{i \in I} (k + b_i) = t \tag{2}$$

Since

$$\begin{aligned} \prod_{i \in I} (k + b_i) &= \prod_{i \in I} (k + a_i - k) \\ &= \prod_{i \in I} a_i \end{aligned}$$

we conclude that $q(k) = t$ if and only if $\prod_{i \in I} a_i = t$. This is true if and only if $\langle a_1, \dots, a_n, t \rangle$ is a YES instance of SUBSET-PRODUCT. \square

Remark 3. It is easy to check (directly from the proof of Theorem 1) that the *k*-FACTOR problem remains NP-complete in the strong sense even if we restrict the set of input polynomials to monic polynomials with all integer roots.

Definition 5 (SUM-OF-COEFFICIENTS problem). The SUM-OF-COEFFICIENTS problem is defined as follows.

Input: $p(x) \in \mathbb{Z}[x]$ and $s \in \mathbb{Z}$

Output: - YES if $p(x)$ has a factor $q(x) \in \mathbb{Z}[x]$ such that the sum of all the coefficients of $q(x)$ is equal to s ,
 - NO otherwise.

Corollary 1. The SUM-OF-COEFFICIENTS problem is NP-complete in the strong sense.

Proof. We prove this result as a Corollary of Theorem 1. We reduce the k -FACTOR problem with $k = 1$ (NP-complete in the strong sense by Theorem 1) to the SUM-OF-COEFFICIENTS problem.

Let $q(x)$ be any factor of $p(x)$. Since $q(1)$ is equal to the sum of all the coefficients of $q(x)$, we conclude that $q(1) = h$ if and only if the sum of all the coefficients of $q(x)$ is equal to h . \square

Definition 6. The CONSTANT-TERM problem is defined as follows.

Input: $p(x) \in \mathbb{Z}[x]$ and $t \in \mathbb{Z}$

Output: - YES if $p(x)$ has a factor $q(x) \in \mathbb{Z}[x]$ such that the constant term of $q(x)$ is equal to t ,
 - NO otherwise.

Corollary 2. The CONSTANT-TERM problem is NP-complete in the strong sense.

Proof. We prove this result as a Corollary of Theorem 1. We reduce the k -FACTOR problem with $k = 0$ (NP-complete in the strong sense by Theorem 1) to the CONSTANT-TERM problem.

Let $q(x)$ be any factor of $p(x)$. Since $q(0)$ is equal to the constant term of $q(x)$, we conclude that $q(0) = h$ if and only if the constant term of $q(x)$ is equal to h . \square

Definition 7 (k -EQUAL-FACTOR problem). Let $k \in \mathbb{Z}$ be any fixed integer. The problem k -EQUAL-FACTOR is defined as follows.

Input: $p(x) \in \mathbb{Z}[x]$

Output: - YES if $p(x)$ has two factors $q(x), r(x) \in \mathbb{Z}[x]$ such that
 $p(x) = q(x) \cdot r(x)$ and $q(k) = r(k)$
 - NO otherwise.

Theorem 2. k -EQUAL-FACTOR is NP-complete in the strong sense.

Proof. Let $k \in \mathbb{Z}$ be any fixed integer. We reduce the PRODUCT-PARTITION problem (Definition 3) to the k -EQUAL-FACTOR problem.

Let $\langle a_1, \dots, a_n \rangle$ be any instance of product partition. Let $b_i = a_i - k$ for $i = 1, \dots, n$ and

$$p(x) = \prod_{i=1}^n (x + b_i)$$

We now prove that $\langle p(x) \rangle$ is a YES instance for k -EQUAL-FACTOR if and only if $\langle a_1, \dots, a_n \rangle$ is a YES instance for PRODUCT-PARTITION.

$\langle p(x) \rangle$ is a YES instance of k -EQUAL-FACTOR if and only if $p(x)$ has two factors $q(x), r(x) \in \mathbb{Z}[x]$ such that $p(x) = q(x) \cdot r(x)$ and $q(k) = r(k)$. Or, equivalently, by Observation 1, if and only if the set $\{1, \dots, n\}$ can be partitioned into two nonempty subsets I and J such that

$$\prod_{i \in I} (k + b_i) = \prod_{j \in J} (k + b_j) \tag{3}$$

Since $b_i = a_i - k$ for $i = 1, \dots, n$, Equation 3 can be rewritten as follows:

$$\prod_{i \in I} (k + a_i - k) = \prod_{j \in J} (k + a_j - k) \tag{4}$$

and then

$$\prod_{i \in I} a_i = \prod_{j \in J} a_j \tag{5}$$

Equation (5) holds if and only if $\langle a_1, \dots, a_n \rangle$ is a YES instance for PRODUCT-PARTITION. \square

Definition 8 (EQUAL-SUM-OF-COEFFICIENTS problem). *The EQUAL-SUM-OF-COEFFICIENTS problem is defined as follows.*

Input: $p(x) \in \mathbb{Z}[x]$

Output: - YES if there exist $q(x), r(x) \in \mathbb{Z}[x]$ such that $q(x) \cdot r(x) = p(x)$ and the sum of all the coefficients of $q(x)$ is equal to the sum of all the coefficients of $r(x)$,
 - NO otherwise.

Corollary 3. *The EQUAL-SUM-OF-COEFFICIENTS problem is strongly NP-complete.*

Proof. Since the sum of the coefficients of any polynomial $p(x)$ is equal to $p(1)$, the proof of this Corollary follows from Theorem 2 setting $k = 1$. \square

Definition 9 (EQUAL-CONSTANT-TERM problem). *The EQUAL-CONSTANT-TERM problem is defined as follows.*

Input: $p(x) \in \mathbb{Z}[x]$

Output: - YES if there exist $q(x), r(x) \in \mathbb{Z}[x]$ such that $q(x) \cdot r(x) = p(x)$ and the constant term of $q(x)$ is equal to the constant term of $r(x)$,
 - NO otherwise.

Corollary 4. *The EQUAL-CONSTANT-TERM problem is strongly NP-complete.*

Proof. Since the constant term of any polynomial $p(x)$ is equal to $p(0)$, the proof of this Corollary follows from Theorem 2 setting $k = 0$. \square

4. Open Questions

4.1. Natural Factors Detection Problem

Let \mathbb{N} denote the set of natural numbers (non-negative integer numbers) and $\mathbb{N}[x]$ the set of natural polynomials (polynomials with coefficients in \mathbb{N}). $\mathbb{N}[x]$ with the usual sum and product operations is a commutative semiring. In fact, this is the free commutative semiring on a single generator $\{x\}$. For theoretical results regarding natural polynomials, we refer the reader to [15].

Definition 10. *The NATURAL-REDUCIBILITY problem is defined as follows.*

Input: $p(x) \in \mathbb{Z}[x]$

Output: - YES if $p(x)$ has two factors $q(x), r(x) \in \mathbb{N}[x]$ such that $p(x) = q(x) \cdot r(x)$,
 - NO otherwise.

Question 1. *Is the NATURAL-REDUCIBILITY problem NP-complete?*

The following example shows a polynomial that is irreducible when considered as an element of $\mathbb{N}[x]$ and reducible when considered as an element of $\mathbb{Z}[x]$.

Example 1. *Let $p(x) = 1 + x^3$. The complete factorization of $p(x)$ in $\mathbb{Z}[x]$ is $p(x) = (1 + x)(1 - x + x^2)$ while $p(x)$ is irreducible in $\mathbb{N}[x]$.*

In the next example, we show that the prime factorization of integer polynomials in $\mathbb{N}[x]$ is not unique.

Example 2. *Let $p(x) = 1 + x + x^2 + x^3 + x^4 + x^5$. The complete factorization of $p(x)$ in $\mathbb{Z}[x]$ is $p(x) = (1 + x)(1 - x + x^2)(1 + x + x^2)$. Since $(1 + x)(1 - x + x^2) \in \mathbb{N}[x]$ and $(1 - x + x^2)(1 + x + x^2) \in \mathbb{N}[x]$, then we have two distinct factorizations of $p(x)$ in $\mathbb{N}[x]$.*

$$\begin{aligned} p(x) &= (1 + x)(1 + x^2 + x^4) \\ &= (1 + x^3)(1 + x + x^2) \end{aligned}$$

Our conjecture is that the NATURAL-REDUCIBILITY problem is NP-complete, but we have not been able to prove it.

4.2. Factors with Specific Coefficients Detection Problem

Let $p(x) = c_0 + c_1x + \dots + c_nx^n$ be any integer polynomial. We denote by $\text{coef}(p(x), m)$, $0 \leq m \leq n$, the coefficient c_m . For values of m outside the interval $[0, n]$, $\text{coef}(p(x), m)$ is equal to 0. According to this definition, $\text{coef}(p(x), n) = c_n$ is the coefficient of the monomial in $p(x)$ with maximum degree (for monic polynomials is always equal to 1) and $\text{coef}(p(x), 0) = c_0$ is the constant term of $p(x)$.

The factor with specific coefficients detection problem is defined as follows.

Definition 11 (FACTOR-WITH-SPECIFIC-COEFFICIENTS). *Let $m \geq 0$ be any fixed integer.*

Input: $p(x) \in \mathbb{Z}[x]$ and $h \in \mathbb{Z}$

Output: - YES if $p(x)$ has a factor $q(x) \in \mathbb{Z}[x]$ such that $\text{coef}(q(x), m) = h$,
 - NO otherwise.

By Corollary 2, we have that for $m = 0$, FACTOR-WITH-SPECIFIC-COEFFICIENTS problem is NP-complete in the strong sense. In fact, when $m = 0$, the problem is equivalent to the CONSTANT-TERM problem.

We now define a problem that is a sort of combination of SUBSET-SUM and SUBSET-PRODUCT problems.

Definition 12 (SUBSET-SUM-OF-PRODUCTS). *Let k be any fixed integer. Given $n + 1$ positive integers a_1, \dots, a_n, t decide whether there exist $I \subseteq \{1, \dots, n\}$ such that*

$$\sum_{\substack{i_1 < i_2 < \dots < i_k \\ i_j \in I, 1 \leq j \leq k}} \left(\prod_{j=1}^k a_{i_j} \right) = t \tag{6}$$

Note that for $k = 1$, the SUBSET-SUM-OF-PRODUCTS problem is nothing but the SUBSET-SUM problem (Definition 1) and then it is NP-complete.

In the following theorem, we prove that the FACTOR-WITH-SPECIFIC-COEFFICIENTS detection problem is not easier than the SUBSET-SUM-OF-PRODUCTS problem.

To this extent we recall the Vieta’s formulas (customized for monic polynomials over the integers with integer roots) that relate the roots of a polynomial to its coefficients.

Theorem 3 (Vieta’s Formulas for monic polynomials over the integers with integer roots). *Let*

$$\begin{aligned} p(x) &= \prod_{i=1}^n (x + a_i) \\ &= x^n + c_{n-1}x^{n-1} + \dots + c_1x + c_0 \end{aligned}$$

with $a_i \in \mathbb{Z}$. We have

$$\forall k \in \{1, \dots, n\} : c_{n-k} = \sum_{1 \leq i_1 < \dots < i_k \leq n} \left(\prod_{j=1}^k a_{i_j} \right)$$

Example 3. *Let*

$$\begin{aligned} P_3(x) &= (x + a_1)(x + a_2)(x + a_3) \\ &= x^3 + (a_1 + a_2 + a_3)x^2 + \\ &\quad (a_1a_2 + a_1a_3 + a_2a_3)x + a_1a_2a_3 \end{aligned}$$

$$\begin{aligned}
 P_4(x) &= (x + a_1)(x + a_2)(x + a_3)(x + a_4) \\
 &= x^4 + (a_1 + a_2 + a_3 + a_4)x^3 + \\
 &\quad (a_1a_2 + a_1a_3 + a_1a_4 + a_2a_3 + a_2a_4 + a_3a_4)x^2 \\
 &\quad + (a_1a_2a_3 + a_1a_2a_4 + a_1a_3a_4 + a_2a_3a_4)x + \\
 &\quad a_1a_2a_3a_4
 \end{aligned}$$

Theorem 4. *The SUBSET-SUM-OF-PRODUCTS problem is reducible to the FACTOR-WITH-SPECIFIC-COEFFICIENTS detection problem.*

Proof. Let $\langle a_1, \dots, a_n, t, k \rangle$ be any instance of SUBSET-SUM-OF-PRODUCTS. Let $m = k, h = t$ and

$$p(x) = \prod_{i=1}^n (x + a_i)$$

Let $\langle p(x), t \rangle$ be the corresponding instance of FACTOR-WITH-SPECIFIC-COEFFICIENTS. We now prove that $\langle a_1, \dots, a_n, t, k \rangle$ is a YES instance of SUBSET-SUM-OF-PRODUCTS if and only if $\langle p(x), t \rangle$ is a YES instance of FACTOR-WITH-SPECIFIC-COEFFICIENTS.

$\langle p(x), t \rangle$ is a YES instance of FACTOR-WITH-SPECIFIC-COEFFICIENTS if and only if there exists a factor $q(x) \in \mathbb{Z}[x]$ of $p(x)$ such that $\text{coef}(q(x), m) = h$. By Observation 1, any factor of $p(x)$ has the form

$$q(x) = \prod_{j=1}^{\text{deg}(q(x))} (x + b_{i_j}) \tag{7}$$

where $I = \{i_1, \dots, i_{\text{deg}(q(x))}\}$ is a suitable subset of $\{1, \dots, n\}$.

By Vieta’s formulas, we know that $\text{coef}(q(x), m)$ can be written as

$$\sum_{\substack{i_1 < i_2 < \dots < i_m \\ i_j \in I, 1 \leq j \leq m}} \left(\prod_{j=1}^m b_{i_j} \right) \tag{8}$$

This ends the proof. \square

We end this section with the following open question.

Question 2. *For which values of m (other than the case $m = 0$) is the FACTOR-WITH-SPECIFIC-COEFFICIENTS problem NP-complete?*

5. Conclusions and Further Works

In this paper, we have introduced and analyzed the computational complexity of the problem of detecting the existence of factors of integer polynomials satisfying specific additional constraints. Even if detecting the existence of factors of an integer polynomial can be accomplished in polynomial time, it turns out that adding simple constraints on factors leads to hard to solve variants of the problem. We prove that problems Q1 to Q6 listed at the end of Section 1 are NP-complete in the strong sense.

Section 4 provides some ideas for further works. In particular, we were surprised that Question 1 (to our knowledge) had not already been previously addressed in the literature since the problem of factoring an integer polynomial over \mathbb{N} instead of over \mathbb{Z} seems to us a very natural and interesting question to investigate.

Author Contributions: Investigation, A.D., E.F. and L.M.; Resources, E.F. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Laboratoire I3S of Université Côte d’Azur.

Data Availability Statement: No data is used or generated.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hoeij, M.V. Factoring polynomials and the knapsack problem. *J. Number Theory* **2002**, *95*, 167–189. [[CrossRef](#)]
2. Lenstra, A.K.; Lenstra, H.W.; Lovasz, L. Factoring polynomials with rational coefficients. *Math. Ann.* **1982**, *261*, 515–534. [[CrossRef](#)]
3. Kaltofen, E. Polynomial factorization 1987–1991. In Proceedings of the LATIN '92, 1st Latin American Symposium on Theoretical Informatics, São Paulo, Brazil, 6–10 April 1992; Simon, I., Ed.; Proceedings; Lecture Notes in Computer Science; Springer: Berlin, Germany, 1992; Volume 583, pp. 294–313. [[CrossRef](#)]
4. Schubert, F. De Inventione Divisorum. *Nova Acta Acad. Sci. Petropolitanae* **1793**, *11*, 172–182.
5. Plaisted, D.A. Sparse Complex Polynomials and Polynomial Reducibility. *J. Comput. Syst. Sci.* **1977**, *14*, 210–221. [[CrossRef](#)]
6. Karpinski, M.; Shparlinski, I.E. On the Computational Hardness of Testing Square-Freeness of Sparse Polynomials. In Proceedings of the AAEECC, Honolulu, HI, USA, 15–19 November 1999; Fossorier, M.P.C., Imai, H., Lin, S., Poli, A., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Germany, 1999; Volume 1719, pp. 492–497.
7. Kaltofen, E.; Koiran, P. On the complexity of factoring bivariate supersparse (Lacunary) polynomials. In Proceedings of the ISSAC, Beijing, China, 24–27 July 2005; Kauers, M., Ed.; pp. 208–215.
8. Grenet, B. Bounded-degree factors of lacunary multivariate polynomials. *J. Symb. Comput.* **2016**, *75*, 171–192. [[CrossRef](#)]
9. Cucker, F.; Koiran, P.; Smale, S. A Polynomial Time Algorithm for Diophantine Equations in One Variable. *J. Symb. Comput.* **1999**, *27*, 21–29. [[CrossRef](#)]
10. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*; W. H. Freeman: New York, NY, USA, 1979.
11. Karp, R. Reducibility among combinatorial problems. In *Complexity of Computer Computations*; Miller, R., Thatcher, J., Eds.; Plenum Press: New York, NY, USA, 1972; pp. 85–103.
12. Yao, A.C.C. *Technical Report CS-TR-1978-662*; New Algorithms in Bin Packing; Department of Computer Science, Stanford University: Stanford, CA, USA, 1978.
13. Ng, C.; Barketau, M.; Cheng, T.; Kovalyov, M.Y. Product partition and related problems of scheduling and systems reliability: computational complexity and approximation. *Eur. J. Oper. Res.* **2010**, *207*, 601–604. [[CrossRef](#)]
14. Garey, M.R.; Johnson, D.S. “Strong” NP-Completeness results: Motivation, examples, and Implications. *J. ACM* **1978**, *25*, 499–508. [[CrossRef](#)]
15. Campanini, F.; Facchini, A. Factorizations of polynomials with integral non-negative coefficients. *Semigroup Forum* **2019**, *99*, 317–332. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.