

## Research Article

# A Non-NP-Complete Algorithm for a Quasi-Fixed Polynomial Problem

Yi-Chou Chen<sup>1</sup> and Hang-Chin Lai<sup>2</sup>

<sup>1</sup> Department of General Education, National Army Academy, Taoyuan 320, Taiwan

<sup>2</sup> Department of Mathematics, National Tsing Hua University, Hsinchu 300, Taiwan

Correspondence should be addressed to Hang-Chin Lai; laihc@mx.nthu.edu.tw

Received 15 January 2013; Accepted 24 February 2013

Academic Editor: Jen-Chih Yao

Copyright © 2013 Y.-C. Chen and H.-C. Lai. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Let  $F : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  be a real-valued polynomial function of the form  $F(x, y) = \sum_{i=0}^s f_i(x)y^i$ , with degree of  $y$  in  $F(x, y) = s \geq 1$ ,  $x \in \mathbb{R}$ . An irreducible real-valued polynomial function  $p(x)$  and a nonnegative integer  $m$  are given to find a polynomial function  $y(x) \in \mathbb{R}[x]$  satisfying the following expression:  $F(x, y(x)) = cp^m(x)$  for some constant  $c \in \mathbb{R}$ . The constant  $c$  is dependent on the solution  $y(x)$ , namely, a quasi-fixed (polynomial) solution of the polynomial-like equation (\*). In this paper, we will provide a non-NP-complete algorithm to solve all quasi-fixed solutions if the equation (\*) has only a finite number of quasi-fixed solutions.

## 1. Introduction and Preliminaries

Lenstra [1] found that a polynomial function  $F(x, y) \in \mathbb{Q}(\alpha)[x, y]$  (where  $\alpha$  is an algebraic number) can be solved to a polynomial function  $y = y(x) \in \mathbb{Q}(\alpha)[x]$  such that  $F(x, y(x)) = 0$ . It can thus be derived to find a polynomial  $y = y(x)$  such that there exists an  $x \in \mathbb{Q}(\alpha)[x]$  as a fixed point of the polynomial function  $F(x, y(x))$ ; that is,

$$F(x, y(x)) = x \quad (1)$$

has a polynomial solution  $y(x) \in \mathbb{Q}(\alpha)[x]$ .

Furthermore, Tung [2, 3] extended (1) to solve  $y(x)$  for the following equation:

$$F(x, y(x)) = cx^m, \quad \text{for } x \in \mathbb{K} \text{ (field)}, \quad m \in \mathbb{N}, \quad (2)$$

where  $c$  is a real constant depending on the polynomial solution  $y(x)$  and the given positive integer  $m \in \mathbb{N}$ .

Recently, Lai and Chen [4] extended the expression (2) to solve  $y(x)$  to satisfy the polynomial equation as the following form:

$$F(x, y(x)) = cp^m(x), \quad x \in \mathbb{R}, \quad (3)$$

where  $p(\cdot)$  is an irreducible polynomial in  $x \in \mathbb{R}$  and the polynomial functions  $F(x, y) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  are written by

$$F(x, y) = \sum_{i=0}^s f_i(x)y^i, \quad (4)$$

with degree  $y$  in  $F(x, y)$  denoted by  $\deg_y F = s > 1$ .

*Remark 1.* Recall that a polynomial function  $y = y(x)$  satisfying (3) is called a quasi-fixed (polynomial) solution corresponding to the real value  $a$ . This number  $a$  is called a quasi-fixed (polynomial) value corresponding to the polynomial solutions  $y = y(x)$ .

Note that the equation may not be solvable; if it is solvable, the number of all solutions may be infinitely many, or finitely many quasi-fixed solutions. If this equation has infinitely many quasi-fixed solutions, by [4, Theorem 3.5], we have

- (1)  $f_s(x)$  must be  $ap^r(x)$  for some  $a \in \mathbb{R}, r \in \mathbb{N}$ ,
- (2) those solutions must assume a fixed form like

$$-\frac{f_{s-1}(x)}{sf_s(x)} + \lambda p^k(x) \quad \text{for any } \lambda \in \mathbb{R}. \quad (5)$$

Moreover, if this equation has only finitely many quasi-fixed solutions, by [4, Corollary 3.3], the bound of all solutions is at most  $\deg_y F + 2$ .

In this paper, we deal with the case of finitely many solutions and provide a non-NP-complete algorithm to obtain all representations of quasi-fixed solutions.

*Remark 2.* An algorithm is called “NP-complete” if the algorithm is computed in an exponential time of the input size “ $n$ ”; that is, the computing time in the algorithm does not exceed  $O(e^n)$ . Otherwise, the algorithm is called “non-NP-complete”; that is, the algorithm can be computed in a polynomial time not exceeding  $O(n^k)$  for some fixed real number  $k$ .

The remainder of the paper is organized as follows: Section 2 introduces an example for the practical algorithm. Section 3 describes the main algorithm and its processes. Section 4 proves that the main algorithm is indeed non-NP-complete. We have given two example problems in Section 5.

### 2. Relative Algorithm and Some Examples

Let  $F(x, y) \in \mathbb{R}[x, y]$  and an irreducible polynomial function  $p(x) \in \mathbb{R}[x]$ . To complete our algorithm, we need an existing algorithm. In [2], Tung has established the following algorithm.

*Algorithm 3* (Tung [5]). Consider the following technique.

*Input.* Given is a polynomial  $F(x, y) \in \mathbb{R}[x, y]$ .

*Output.* All solutions  $y = y(x)$  satisfy

$$F(x, y(x)) = a, \quad \text{for some } a \in \mathbb{R}, \quad (6)$$

in polynomial time of  $w\mu$ , where  $w$  is the computer memory of all coefficients of  $F(x, y)$  and  $\mu = \deg_x F(x, y) + \deg_y F(x, y)$ . Moreover, the number of all solutions is at most  $\deg_y F(x, y)$ .

We provide a simple example to explain our aim for solving the quasi-fixed polynomial solutions related to Algorithm 5 (in Section 3) as follows.

*Example 4.* Let

$$F(x, y) = p(x)[y^2 - p^2(x) - 4] \quad (7)$$

and  $p(x) = x^2 + x + 1$ . Then  $\deg_y F = s = 2$ . Solve all quasi-fixed solutions of

$$F(x, y) = ap^2(x). \quad (8)$$

By (7) and (8), we obtain

$$p(x)[y^2 - p^2(x) - 4] = ap^2(x), \quad (9)$$

divides both sides by  $p(x)$ , then it becomes

$$y^2 - p^2(x) - 4 = ap(x). \quad (10)$$

It follows that

$$y^2 - 4 = 0 \pmod{p(x)}, \quad (11)$$

and we have

$$y = \begin{cases} zp(x) + 2 \\ zp(x) - 2 \end{cases} = \begin{cases} 2 \pmod{p(x)} \\ -2 \pmod{p(x)} \end{cases}, \quad (12)$$

with an indeterminate  $z$ . To determine  $z$ , we substitute this  $y$  (in (12)) into  $F(x, y)$ , then (8) becomes

$$(P_1) \begin{cases} (1) & F(x, y) = F(x, zp(x) + 2) = ap^2(x) \\ (2) & F(x, y) = F(x, zp(x) - 2) = ap^2(x). \end{cases} \quad (13)$$

This yields

$$(P_2) \begin{cases} (1) & p^2(x)[p(x)z^2 + 4z - p(x)] = ap^2(x) \\ (2) & p^2(x)[p(x)z^2 - 4z - p(x)] = ap^2(x). \end{cases} \quad (14)$$

Both sides of (14) divide by  $p^2(x)$  to become

$$(P_3) \begin{cases} (1) & p(x)z^2 + 4z - p(x) = a \\ (2) & p(x)z^2 - 4z - p(x) = a. \end{cases} \quad (15)$$

According to Algorithm 3, we obtain  $z = 1$  and  $z = -1$ , so the solutions (12) of problem (P) are

$$y = \begin{cases} zp(x) + 2 \\ zp(x) - 2 \end{cases} = \begin{cases} p(x) + 2 \\ p(x) - 2 \end{cases} \quad \text{or} \quad \begin{cases} -p(x) + 2 \\ -p(x) - 2. \end{cases} \quad (16)$$

In Section 3, we provide a non-NP-complete algorithm to satisfy (3).

### 3. Computing Procedure

In this paper, we may assume that the number of all solutions of (3) is finite and then constitute an algorithm of the approximate solutions for the quasi-fixed polynomial equation (3) as follows.

*Algorithm 5.* Consider the following technique.

*Input.* Given a polynomial  $F(x, y) \in \mathbb{R}[x, y]$ ,  $p(x) \in \mathbb{R}[x]$  irreducible and  $m \in \mathbb{N}$ .

*Output.* All solutions  $y = y(x)$  satisfy

$$F(x, y(x)) = ap^m(x) \quad \text{for some } a \in \mathbb{R}, \quad (17)$$

by a non-NP-complete algorithm.

The following definitions are given by [6, page 421].

*Definition 6.* (i) The content of  $F(x, y)$  in (4), denoted by  $\text{cont}_y F$ , is defined by the greatest common divisor (g.c.d):

$$\text{gcd}(f_s(x), f_{s-1}(x), \dots, f_0(x)) = \text{cont}_y F. \quad (18)$$

(ii) The primitive part  $\text{pp}_y F$  of  $F(x, y)$  is  $F(x, y)/\text{cont}_y F \in \mathbb{R}[x, y]$ , and  $F(x, y)$  is primitive if  $\text{cont}_y F = 1$ .

From (i) and (ii), we have

$$F(x, y) = \text{cont}_y F \cdot \text{pp}_y F. \quad (19)$$

To complete the following algorithm, we need the following elementary property.

**Lemma 7.** Let  $F(x, y) \in \mathbb{R}[x, y]$  and  $p(x)$  be an irreducible polynomial in  $\mathbb{R}[x]$ . Consider the module equation

$$F(x, y) = 0 \pmod{p(x)}. \quad (20)$$

The number of all solutions  $y = y(x) \pmod{p(x)}$  is thus at most  $\deg_y F$ .

*Assumption.* Throughout this algorithm, for any  $F(x, y) \in \mathbb{R}[x, y]$ , one can solve all solutions  $y = g(x) \pmod{p(x)}$  of the equation

$$F(x, g(x)) = 0 \pmod{p(x)}. \quad (21)$$

The procedure of our main algorithm is described below.

*Procedures of Main Algorithm (Algorithm 5)*

*Step 0.* If  $\text{cont}_y F \mid p^m(x)$  and  $\text{cont}_y F = 1$ , then let  $F_1(x, y) = F(x, y)$  and move to (Step 1, (1.2.2)).

*Step 1.* For convenience, we let

$$F(x, y) = (\text{cont}_y F) F_1(x, y), \quad (22)$$

where  $F_1(x, y) \in \mathbb{R}[x, y]$  is a primitive polynomial.

*Step 1.1.* If  $\text{cont}_y F \nmid p^m(x)$ , we would have the problem

$$F(x, y) = cp^m(x), \quad (23)$$

to deduce that  $\text{cont}_y F \mid c \cdot p^m(x)$ . But  $\text{cont}_y F \nmid p^m(x)$ , then  $c = 0$ . Consequently, (23) becomes

$$F(x, y) = 0. \quad (24)$$

We can then solve all solutions  $y(x)$  for  $F(x, y) = 0$  to get a solution set

$$Y_0 = \{y(x) : F(x, y(x)) = 0\}. \quad (25)$$

*Step 1.2.* If  $\text{cont}_y F \mid p^m(x)$  and  $\text{cont}_y F \neq 1$ , then  $\text{cont}_y F = p^{\ell_1}(x)$  with  $\ell_1 \leq m$ . In this case, (23) becomes

$$F_1(x, y) = cp^{m-\ell_1}(x). \quad (26)$$

*Step 1.2.1.* In case  $\ell_1 = m$ , then (26) becomes  $F_1(x, y) = c$  which can be solved by Algorithm 3 to obtain all solutions for the equation  $F_1(x, y) = c$  to get  $y = y(x)$  and obtain a set

$$W_0 = \{y(x) : F_1(x, y(x)) = c\}. \quad (27)$$

*Step 1.2.2.* If  $\ell_1 < m$ , then  $m - \ell_1 > 0$  and we can divide both sides of (26) by  $p(x)$ ; consequently,

$$F_1(x, y) = 0 \pmod{p(x)}. \quad (28)$$

According to Lemma 7, the solution number does not exceed  $\deg_y F$ , thus we may assume that  $y = a_0(x)$  is a solution of (28) with  $\deg a_0(x) < \deg p(x)$ ; please note that the choice of  $a_0(x)$  may be larger than 1 and we may define a solution set  $T_0(F)$  which collects such  $a_0(x)$  by setting as the following form:

$$a_0(x) \in T_0(F) = \{a_1^0(x), a_2^0(x), \dots, a_{r_0}^0(x)\}. \quad (29)$$

Consequently, the expression (23) can go to Step 2.

*Step 2.* For each  $a_0(x) \in T_0(F)$  in (Step 1, (1.2.2)), we can replace  $y$  by  $zp(x) + a_0(x)$  in (23), and then (23) becomes

$$F(x, zp(x) + a_0(x)) = cp^m(x). \quad (30)$$

We let  $H_1(x, z) = F(x, zp(x) + a_0(x))$ . Thus it follows from Definition 6(ii) that there exists

$$F_2(x, y) = \text{pp}_y H_1 \in \mathbb{R}[x, y] \quad (31)$$

which is a primitive polynomial function so that

$$H_1(x, y) = (\text{cont}_y H_1) F_2(x, y). \quad (32)$$

Equation (30) then becomes

$$H_1(x, y) = cp^m(x). \quad (33)$$

Next, we will prove that  $p(x) \mid \text{cont}_y H_1$ .

*Step 2.1.* If  $\text{cont}_y H_1 \nmid p^m(x)$ , (32) and (33) can be extrapolated as to  $\text{cont}_y H_1 \mid c \cdot p^m(x)$ . Since  $\text{cont}_y H_1 \nmid p^m(x)$ , then  $c = 0$ . Consequently, (33) becomes

$$H_1(x, y) = 0. \quad (34)$$

We can then solve all solutions  $y(x)$  for  $H_1(x, y) = 0$  to get a solution set

$$Y_1 = \{y(x) p(x) + a_0(x) : H_1(x, y(x)) = 0, a_0(x) \in T_0(F)\}. \quad (35)$$

*Step 2.2.* If  $\text{cont}_y H_1 \mid p^m(x)$ , then  $\text{cont}_y H_1 = p^{\ell_2}(x)$  with  $\ell_2 \leq m$ . In this case, (33) becomes

$$F_2(x, y) = cp^{m-\ell_2}(x). \quad (36)$$

*Step 2.2.1.* In case  $\ell_2 = m$ , (36) becomes

$$F_2(x, y) = c \quad (37)$$

which can be solved by Algorithm 3 to get  $y = y(x)$  and obtain a set

$$W_1 = \{y(x) p(x) + a_0(x) : F_2(x, y(x)) = c, a_0(x) \in T_0(F)\}. \quad (38)$$

*Step 2.2.2.* If  $\ell_2 < m$ , then  $m - \ell_2 < 0$  and we can divide both sides of (36) by  $p(x)$ ; consequently,

$$F_2(x, y) = 0 \pmod{p(x)}. \quad (39)$$

By the same reasoning used in [Step 1.2.2], we can solve  $y = a_1(x)$  with  $\deg a_1(x) < \deg p(x)$  and obtain the set

$$a_1(x) \in V_1 = \{a_1^1(x), a_2^1(x), \dots, a_{r_1}^1(x)\}. \quad (40)$$

We let

$$T_1(F) = \{a_1(x)p(x) + a_0(x) : a_1(x) \in V_1, a_0(x) \in T_0(F)\}, \quad (41)$$

Consequently, expression (33) can go to Step 3.

*Step 3.* For each  $a_1(x)p(x) + a_0(x) \in T_1(F)$ , we can replace  $y$  by  $zp^2(x) + a_1(x)p(x) + a_0(x)$  in (23). It then becomes

$$F(x, zp^2(x) + a_1(x)p(x) + a_0(x)) = cp^m(x). \quad (42)$$

Moreover, we let  $H_2(x, z) = F(x, zp^2(x) + a_1(x)p(x) + a_0(x))$  and

$$H_2(x, y) = (\text{cont}_y H_2) F_3(x, y) \quad (43)$$

for a primitive polynomial  $F_3(x, y) = \text{pp}_y H_2 \in \mathbb{R}[x, y]$ , so that (42) becomes

$$H_2(x, y) = cp^m(x). \quad (44)$$

Next, we will prove that  $p^2(x) \mid \text{cont}_y H_2$ .

*Step 3.1.* If  $\text{cont}_y H_2 \nmid p^m(x)$ , and (43) and (44) are extrapolated, we have

$$\text{cont}_y H_2 \mid c \cdot p^m(x). \quad (45)$$

Since  $\text{cont}_y H_2 \nmid p^m(x)$ , we have  $c = 0$ .

Thus (44) becomes

$$H_2(x, y) = 0. \quad (46)$$

Hence, we solve all solutions  $y(x)$  such that  $H_2(x, y) = 0$  to obtain a solution set

$$Y_2 = \{y(x)p^2(x) + a_1(x)p(x) + a_0(x) : H_2(x, y(x)) = 0, \\ a_1(x)p(x) + a_0(x) \in T_1(F)\}. \quad (47)$$

*Step 3.2.* If  $\text{cont}_y H_2 \mid p^m(x)$ , then  $\text{cont}_y H_2 = p^{\ell_3}(x)$  for some positive integer  $\ell_3 \leq m$  and (44) becomes

$$F_3(x, y) = cp^{m-\ell_3}(x). \quad (48)$$

*Step 3.2.1.* If  $\ell_3 = m$ , then (48) becomes

$$F_3(x, y) = c \quad (49)$$

and, by Algorithm 3, we compute  $y = y(x)$  and collect

$$y(x)p^2(x) + a_1(x)p(x) + a_0(x) \quad (50)$$

in  $W_2$  for all  $a_1(x)p(x) + a_0(x) \in T_1(F)$ . Hence, we output

$$W_2 = \{y(x)p^2(x) + a_1(x)p(x) + a_0(x) : F_3(x, y(x)) = c, \\ a_1(x)p(x) + a_0(x) \in T_1(F)\}. \quad (51)$$

*Step 3.2.2.* If  $\ell_3 < m$ , we use  $p(x)$  to divide both sides of (48) to obtain

$$F_3(x, y) = 0 \pmod{p(x)}. \quad (52)$$

By the same reasoning used in (Step 1.2.2), we can solve  $y = a_2(x)$  with  $\deg a_2(x) < \deg p(x)$  and obtain the set

$$a_2(x) \in V_2 = \{a_1^2(x), a_2^2(x), \dots, a_{r_2}^2(x)\}. \quad (53)$$

We let

$$T_2(F) = \{a_2(x)p^2(x) + a_1(x)p(x) + a_0(x) \\ : a_2(x) \in V_2, a_1(x)p(x) + a_0(x) \in T_1(F)\}, \quad (54)$$

thus expression (44) can go to Step 4.

Continuing this process, we get the  $(k-1)$ -th step and a sequence  $\{\ell_1, \ell_2, \dots, \ell_{k-1}\}$  and go to the next  $k$ -th step.

*Step k.* For each  $\sum_{i=0}^{k-2} a_i(x)p^i(x) \in T_{k-2}(F)$ , we can replace  $y$  by  $zp^{k-1}(x) + \dots + a_0(x)$  in (23), and then (23) becomes

$$F(x, zp^{k-1}(x) + \dots + a_0(x)) = cp^m(x). \quad (55)$$

We let  $H_{k-1}(x, z) = F(x, zp^{k-1}(x) + \dots + a_0(x))$  and

$$H_{k-1}(x, y) = (\text{cont}_y H_{k-1}) F_k(x, y) \quad (56)$$

for a primitive polynomial  $F_k(x, y) \in \mathbb{R}[x, y]$ , then (55) becomes

$$H_{k-1}(x, y) = cp^m(x). \quad (57)$$

Note that in Section 4, Lemma 9, we will prove that  $p^{k-1}(x) \mid \text{cont}_y H_{k-1}$ .

*Step k.1.* If  $\text{cont}_y H_{k-1} \nmid p^m(x)$ , by (56) and (57), we have

$$\text{cont}_y H_{k-1} \mid c \cdot p^m(x), \quad (58)$$

since  $\text{cont}_y H_{k-1} \nmid p^m(x)$ , we have  $c = 0$ .

Thus (57) becomes

$$H_{k-1}(x, y) = 0. \tag{59}$$

Hence, we solve all solutions  $y(x)$  such that  $H_{k-1}(x, y) = 0$  to obtain a solution set

$$Y_{k-1} = \left\{ y(x) p^{k-1} + \sum_{i=0}^{k-2} a_i(x) p^i(x) : H_{k-1}(x, y(x)) = 0, \right. \\ \left. \sum_{i=0}^{k-2} a_i(x) p^i(x) \in T_{k-2}(F) \right\}. \tag{60}$$

*Step k.2.* If  $\text{cont}_y H_{k-1} \mid p^m(x)$ , then  $\text{cont}_y H_{k-1} = p^{\ell_k}(x)$  for some positive integer  $\ell_k \leq m$  and (57) becomes

$$F_k(x, y) = cp^{m-\ell_k}(x). \tag{61}$$

*Step k.2.1.* If  $\ell_k = m$ , then (61) becomes

$$F_k(x, y) = c. \tag{62}$$

By Algorithm 3, we compute  $y = y(x)$  and obtain

$$y(x) p^{k-1}(x) + a_{k-2}(x) p^{k-2}(x) + \dots + a_0(x) \tag{63}$$

in  $W_{k-1}$ , for all  $a_{k-2}(x) p^{k-2}(x) + \dots + a_0(x) \in T_{k-2}(F)$ . Hence, we output

$$W_{k-1} = \left\{ y(x) p^{k-1}(x) + \dots + a_0(x) : F_k(x, y(x)) = c, \right. \\ \left. \sum_{i=0}^{k-2} a_i(x) p^i(x) \in T_{k-2}(F) \right\}. \tag{64}$$

*Step k.2.2.* If  $\ell_k < m$ , we use  $p(x)$  to divide both sides of (61) to obtain

$$F_k(x, y) = 0 \pmod{p(x)}. \tag{65}$$

By the same reasoning used in (Step 1.2.2), we can solve  $y = a_{k-1}(x)$  with  $\text{deg } a_{k-1}(x) < \text{deg } p(x)$  and obtain the set

$$a_{k-1}(x) \in V_{k-1} = \{a_1^{k-1}(x), a_2^{k-1}(x), \dots, a_{k-1}^{k-1}(x)\}. \tag{66}$$

We let

$$T_{k-1}(F) = \left\{ \sum_{i=0}^{k-2} a_i(x) p^i(x) : a_{k-1}(x) \in V_{k-1}, \right. \\ \left. \sum_{i=0}^{k-2} a_i(x) p^i(x) \in T_{k-2}(F) \right\}, \tag{67}$$

thus expression (57) can go to the next step.

As explained in Remark 8, this algorithm is not an infinite loop.

*Remark 8.* (i) If  $V_i = \emptyset$ , then  $V_{i+1} = \emptyset$  and if  $V_{i+1} = \emptyset$ , then  $T_{i+1}(F) = \emptyset$  for  $i \geq 0$ . This means that if  $V_i = \emptyset$ , then the work finishes at step  $i + 1$  for  $i \geq 0$ .

(ii) This algorithm is not an infinite loop. Since  $\ell_k < m$ , and by Corollary 13, we have  $\ell_k < \ell_{k+1}$  for  $k = 1, 2, \dots$ , thus this algorithm will terminate at the  $(m + 1)$ -th Step. We can thus say that this algorithm is a finite loop algorithm.

(iii) We may assume that the cardinal number  $|W_i| < \infty$ , for  $i = 1, 2, \dots$ ; however, if  $|W_i| = \infty$ , then Algorithm 5 has an infinite number of solutions, which contradicts our assumption.

Remark 8(ii) shows that the maximum number of steps in this algorithm is  $m + 1$ . Moreover, if the  $(m + 1)$ -th Step happens, then we obtain

$$p^m(x) \mid H_m(x, y) \tag{68}$$

by Corollary 11 and let

$$H_m(x, y) = p^m(x) F_{m+1}(x, y). \tag{69}$$

Thus, the possible solutions of  $H_m(x, y) = cp^m(x)$  are those solutions  $y(x)$  satisfying

$$F_{m+1}(x, y(x)) = c. \tag{70}$$

Moreover, by Algorithm 3, we can get a solution set  $Y$  such that the bound of all solutions is " $s \cdot |T_{m-1}(F)|$ ." So the solution set of (3) is contained in

$$T = \left( \bigcup_{i=0}^{m-1} Y_i \right) \cup \left( \bigcup_{i=0}^{m-1} W_i \right) \cup Y. \tag{71}$$

Finally, we check each element  $u(x) \in T$  to determine whether or not

$$F(x, u(x)) = cp^m(x). \tag{72}$$

The arithmetic computing time to solve each element  $u(x)$  in  $T$  only requires non-NP-complete computing time and to check that each  $u(x) \in T$  satisfies

$$F(x, u(x)) = cp^m(x). \tag{73}$$

Non-NP-complete computing time is also needed given the cardinal number of  $T$ :

$$|T| \leq |W_0| + |Y_0| + |Y| + \sum_{i=0}^{m-2} |W_{i+1}| + \sum_{i=0}^{m-2} |Y_{i+1}| \\ \text{(By Algorithm 3 and Lemma 7)} \tag{74}$$

$$\leq 2s + s \cdot |T_{m-1}(F)| + 2 \sum_{i=0}^{m-2} s \cdot |T_i(F)|,$$

where  $|T_i(F)|$ ,  $|W_i|$ , and  $|Y_i|$  denote the cardinal number of  $T_i(F)$ ,  $W_i$ , and  $Y_i$ . The remaining work "Checks the bound of  $|T_i(F)|$ , for each  $i = 1, 2, \dots, m$ ." In Section 4, Theorem 16, we prove that

$$|T_i(F)| \leq s \tag{75}$$

for  $i = 1, 2, \dots, m - 1$ .

By (74) and (75), we have the cardinal number

$$\begin{aligned} |T| &\leq 2s + s \cdot |T_{m-1}(F)| + 2 \sum_{i=0}^{m-2} s \cdot |T_i(F)| \\ &\leq 2s + s^2 + 2 \sum_{i=0}^{m-2} s^2 \\ &= 2s + (2m-1)s^2. \end{aligned} \quad (76)$$

That is, we have to check at most “ $2s + (2m-1)s^2$ ” solutions  $u(x)$  to satisfy

$$F(x, u(x)) = cp^m(x). \quad (77)$$

Checking each solution only takes non-NP-complete time, thus this algorithm needs non-NP-complete time. That is, Algorithm 5 is indeed a non-NP-complete time algorithm. Note that, for each  $k$ -th Step, whether  $y(x)p^{k-1}(x) + a_{k-2}p^{k-2}(x) + \dots + a_0(x)$  belongs to  $Y_{k-1}$  or  $W_{k-1}$ , the upper bound of all elements in  $T$  may not be as large; in fact,

$$|T| \leq 2s + s^2 + (m-1)s^2 = 2s + ms^2. \quad (78)$$

In the next Section 4, we will complete all necessary properties for this algorithm.

#### 4. Main Theorems

For convenience, we describe some interesting properties of the above algorithm.

**Lemma 9.** Let  $F(x, y) \in \mathbb{R}[x, y]$ ,  $p(x), a(x) \in \mathbb{R}[x]$ , and  $z$  be a parameter, if  $p(x) \mid F(x, a(x))$ , then  $p(x) \mid F(x, zp(x) + a(x))$ .

*Proof.* Let  $F(x, y) = f_s(x)y^s + f_{s-1}(x)y^{s-1} + \dots + f_0(x)$ . Then

$$\begin{aligned} &F(x, zp(x) + a(x)) \\ &= f_s(x)(zp(x) + a(x))^s \\ &\quad + f_{s-1}(x)(zp(x) + a(x))^{s-1} + \dots + f_0(x) \\ &= f_s(x)(z^s p^s(x) + sz^{s-1} p^{s-1} a(x) + \dots + a^s(x)) \\ &\quad + f_{s-1}(x)(z^{s-1} p^{s-1}(x) + (s-1) \\ &\quad\quad z^{s-2} p^{s-2} a(x) + \dots + a^{s-1}(x)) + \dots + f_0(x) \\ &= f_s(x)(Q_s(x, z)p(x) + a^s(x)) + f_{s-1}(x) \\ &\quad \times (Q_{s-1}(x, z)p(x) + a^{s-1}(x)) + \dots + f_0(x) \end{aligned}$$

(where  $Q_i(x, z)$

$$\begin{aligned} &= z^i p^{i-1}(x) + iz^{i-1} p^{i-2}(x) a(x) + \dots + iza^{i-1}(x)) \\ &= p(x)(f_s(x)Q_s(x, z) + f_{s-1}(x)Q_{s-1}(x, z) \\ &\quad + \dots + f_1(x)Q_1(x, z)) \\ &\quad + (f_s(x)a^s(x) + f_{s-1}(x)a^{s-1}(x) + \dots + f_0(x)) \\ &= p(x)(f_s(x)Q_s(x, z) + f_{s-1}(x)Q_{s-1}(x, z) \\ &\quad + \dots + f_1(x)Q_1(x, z)) + F(x, a(x)). \end{aligned} \quad (79)$$

Thus,  $p(x) \mid F(x, a(x)) \Rightarrow p(x) \mid F(x, zp(x) + a(x))$ .  $\square$

The definitions of  $H_i(x, y)$ ,  $\text{cont}_y H_i$ ,  $F_{i+1}(x, y)$ , and  $a_i(x)$  for  $i = 1, 2, \dots, m+1$  in Algorithm 5 are

$$H_k(x, z) = F(x, zp^k(x) + \dots + a_0(x)), \quad (80)$$

$$H_k(x, y) = (\text{cont}_y H_k) F_{k+1}(x, y), \quad (81)$$

for a primitive polynomial  $F_{k+1}(x, y) \in \mathbb{R}[x, y]$  and  $p(x) \mid F_{k+1}(x, a_k(x))$  for  $k = 0, 1, \dots, m$ .

Next, we prove some properties about  $H_k(x, y)$  for  $k = 1, 2, \dots, m+1$ .

**Lemma 10.** Consider  $p(x)\text{cont}_y H_k \mid \text{cont}_y H_{k+1}$ , for  $1 \leq k \leq m$ .

*Proof.* From (80), we have

$$\begin{aligned} H_{k+1}(x, z) &= F(x, zp^{k+1}(x) + a_k(x)p^k(x) + \dots + a_0(x)) \\ &= F(x, (zp(x) + a_k(x))p^k(x) + \dots + a_0(x)) \\ &= H_k(x, zp(x) + a_k(x)) \quad (\text{by (80)}) \\ &= (\text{cont}_z H_k) F_{k+1}(x, zp(x) + a_k(x)) \quad (\text{by (81)}). \end{aligned} \quad (82)$$

By definition of  $a_k(x)$ , we have  $p(x) \mid F_{k+1}(x, a_k(x))$ . From Lemma 9, we obtain  $p(x) \mid F_{k+1}(x, zp(x) + a_k(x))$ . That is,

$$p(x)\text{cont}_y H_k \mid H_{k+1}(x, z), \quad (83)$$

and, by (81), we obtain  $p(x)\text{cont}_y H_k \mid \text{cont}_y H_{k+1}$ .  $\square$

By Lemma 9,  $p(x) \mid H_1(x, z)$ , and the definition of  $\text{cont}_y H_1$ , it follows that  $p(x) \mid \text{cont}_y H_1$ . Moreover, by Lemma 10, we can easily obtain the following corollaries.

**Corollary 11.** Consider  $p^i(x) \mid \text{cont}_y H_i$ , for  $i \geq 1$ .

To complete the main theorem, we give the following definitions.

*Definition 12.* (i) The function  $f(x)$  is said to have  $p(x)$ -power  $\ell$ , denoted by  $\delta_p(f(x)) = \ell$ , if  $p(x)^\ell \mid f(x)$  and  $p(x)^{\ell+1} \nmid f(x)$ .



(ii) We say that  $F(x, y)$  has  $p(x)$ -power  $\ell$ , if  $\text{cont}_y F$  has  $p(x)$ -power  $\ell$ .

The following corollary is an immediate result of Lemma 10 and Definition 12.

**Corollary 13.** Consider  $\ell_i < \ell_{i+1}$ , for  $i \geq 1$ .

*Proof.* Since  $\delta_p(H_i) = \ell_{i+1}$  and by Lemma 10, we have

$$p(x) \text{cont}_y H_k \mid \text{cont}_y H_{k+1}. \quad (84)$$

It follows that

$$1 + \delta_p(H_i) = \delta_p(p(x)H_i) \leq \delta_p(H_{i+1}), \quad (85)$$

so  $1 + \ell_i \leq \ell_{i+1}$ ; hence,  $\ell_i < \ell_{i+1}$ .  $\square$

Next, we obtain some properties of  $p(x)$ -power as follows.

**Lemma 14.** Let  $f(x), g(x), h(x) \in \mathbb{R}[x]$  and  $F(x, y)$  is defined in (4). Then

- (i)  $\delta_p(fg) = \delta_p(f) + \delta_p(g)$ ,
- (ii)  $\delta_p(f + g) \geq \min\{\delta_p(f), \delta_p(g)\}$ , and
- (iii)  $\delta_p(F) = \min_{0 \leq i \leq s} \{\delta_p(f_i)\}$ .
- (iv) If  $h = f + g$  and  $\delta_p(h) < \delta_p(g)$ , then  $\delta_p(h) = \delta_p(f)$ .

For each  $k \in \mathbb{N}$ , we rewrite  $T_k(F)$  defined in the procedures of Algorithm 5 as the following definitions.

*Definition 15.* Let  $F(x, y) \in \mathbb{R}[x, y]$ , an irreducible polynomial  $p(x) \in \mathbb{R}[x]$ , and  $z$  be a parameter. We denote the sets  $T_i(F)$ ,  $i \in \mathbb{N}$  defined in Algorithm 5 as follows:

- (1)  $T_0(F) = \{a(x) \in \mathbb{R}[x] : p(x)\text{cont}_z F(x, z) \mid F(x, a(x)) \text{ with } 0 \leq \deg a(x) < \deg p(x)\}$ ,
- (2)  $T_k(F) = \{b_k(x) \in \mathbb{R}[x] : b_k(x) = a(x)p^k(x) + b_{k-1}(x) \text{ satisfies (D1) and (D2)}\}$  where for  $k \geq 1$ , and the (D1) and (D2) properties are given by the following:

- (D1):  $0 \leq \deg a(x) < \deg p(x)$  and  $b_{k-1}(x) \in T_{k-1}(F)$ ,
- (D2):  $p(x)\text{cont}_z F(x, zp^k(x) + b(x)) \mid F(x, a(x)p^k(x) + b(x))$ .

From the above definitions, we obtain a bound of the cardinal number  $|T_k(F)|$ .

**Theorem 16.** For each  $k \in \mathbb{N}$ , then  $|T_k(F)| \leq \deg_y F$  ( $|\cdot|$  means the cardinal number).

*Proof.* This theorem will be proven by induction on  $\deg_y F$ .

- (i) When  $\deg_y F = 1$ , claim that  $|T_k(F)| \leq 1$  for any  $k \in \mathbb{N}$ . For this purpose, we consider

$$F(x, y) = f_1(x)y + f_0(x) \text{ for some } f_1(x), f_0(x) \in \mathbb{R}[x]. \quad (86)$$

(1) If  $k = 0$ , and let  $v_0 = \delta_p(F(x, z))$  and  $F(x, y) = f_1(x)y + f_0(x) = p^{v_0}(x)(g_1^0(x)y + g_0^0(x))$  for some coprime polynomials  $g_1^0(x), g_0^0(x) \in \mathbb{R}[x]$ .

The only possible solution  $a_0(x)$  with  $0 \leq \deg a_0(x) < \deg p(x)$  and  $p(x)\text{cont}_y F(x, z) \mid F(x, a_0(x))$  is

$$a_0(x) = (g_1^0(x))^{-1} (g_0^0(x)) \text{ mod } p(x). \quad (87)$$

That is,  $|T_0(F)| \leq 1$ .

(2) Next, we assume that  $|T_{k-1}(F)| \leq 1$ , thus we deduce that  $|T_k(F)| \leq 1$ .

Let

$$v_k = \delta_p(F(x, zp^k(x) + b_{k-1}(x))) \quad (88)$$

for some  $b_{k-1}(x) \in T_{k-1}(F)$ ,

and  $F(x, zp^k(x) + b_{k-1}(x)) = p^{v_k}(x)(g_1^k(x)z + g_0^k(x))$  for some  $g_1^k(x), g_0^k(x) \in \mathbb{R}[x]$ . Consider

$$T_k(F) = \left\{ b_k(x) \in \mathbb{R}[x] : b_k(x) = a(x)p^k(x) + b_{k-1}(x) \text{ satisfies (D1) and (D2)} \right\}, \quad (89)$$

and since  $|T_{k-1}(F)| \leq 1$  by assumption,  $b_{k-1}(x)$  is uniquely decided, as is  $v_k$ . The remaining task is to prove that  $a_k$  is also uniquely determined. As  $b_k(x) \in T_k(F)$  with  $b_k(x) = a_k(x)p^k(x) + b_{k-1}(x)$ ,  $a_k(x)$  satisfies

$$0 \leq \deg a_k(x) < \deg p(x) \quad (90)$$

such that

$$p(x) \text{cont}_y F(x, zp^k(x) + b_{k-1}(x)) \mid F(x, a_k(x)p^k(x) + b_{k-1}(x)) \quad (91)$$

for some  $b_{k-1}(x) \in T_{k-1}(F)$ ,

then

$$a_k(x)g_1^k(x) + g_0^k(x) = 0 \text{ mod } p(x), \quad (92)$$

it follows that  $a_k(x)$  must be

$$a_k(x) = (g_1^k(x))^{-1} (g_0^k(x)) \text{ mod } p(x). \quad (93)$$

Consequently,  $|T_k(F)| \leq 1$ .

By (1) and (2), we get that  $|T_k(F)| \leq 1 = \deg_y F$  for any  $k \in \mathbb{N}$ .

(ii) We will prove this theorem by mathematical induction. Assume that

$$|T_k(F_1)| \leq \deg_y F_1, \quad (94)$$

for any  $F_1 \in \mathbb{R}[x]$  with  $\deg_y F_1 \leq n - 1$ , we want to show

$$|T_k(F)| \leq \deg_y F, \quad (95)$$

for any  $F \in \mathbb{R}[x]$  with  $\deg_y F = n$ .

Let  $u = \max\{\delta_p(F(x, c(x))) : c(x) \in T_k(F)\}$  and choose  $d(x) \in T_k(F)$  with

$$\delta_p(F(x, d(x))) = u. \quad (96)$$

Indeed,

$$\delta_p(F(x, d(x))) \geq \delta_p(F(x, c(x))) \quad \text{for any } c(x) \in T_k(F). \quad (97)$$

By the division rule, let  $F(x, y)$  be divided by  $y - d(x)$ , giving us

$$F(x, y) = (y - d(x))F_1(x, y) + F(x, d(x)), \quad (98)$$

for some  $F_1(x, y) \in \mathbb{R}[x, y]$  and  $\deg_y F_1(x, y) = n - 1$ .

We want to show that

$$\text{if } c(x) \in T_k(F) - \{d(x)\}, \text{ then } c(x) \in T_k(F_1). \quad (99)$$

For any  $c(x) \in T_k(F) - \{d(x)\}$ , by the definition of  $T_k(F)$ , we have

$$\begin{aligned} c(x) &= a_1(x)(p(x))^k + b_1(x), \\ d(x) &= a_2(x)(p(x))^k + b_2(x), \end{aligned} \quad (100)$$

for some  $b_1(x), b_2(x) \in T_{k-1}(F)$  and  $0 \leq \deg a_1(x)$ ,  $\deg a_2(x) < \deg p(x)$ . Since the elements  $d(x)$  and  $c(x)$  are distinct, so  $a_1(x) \neq a_2(x)$  or  $b_1(x) \neq b_2(x)$ ,

(a) If  $b_1(x) \neq b_2(x)$ , by the definition of  $b_1(x)$  and  $b_2(x)$ , we have  $\deg b_1(x), \deg b_2(x) < \deg p^k(x)$ , so

$$\delta_p(b_1(x) - b_2(x)) < k \leq \delta_p((a_1(x) - a_2(x))p^k(x)). \quad (101)$$

Therefore, one has the following:

- (i)  $\delta_p(c(x) - d(x)) = \delta_p((a_1(x) - a_2(x))p^k(x) + (b_1(x) - b_2(x))) = \delta_p(b_1(x) - b_2(x))$  (by Lemma 14, (iv)),
- (ii)  $\delta_p(b_1(x) - d(x)) = \delta_p((-a_2(x))p^k(x) + (b_1(x) - b_2(x))) = \delta_p(b_1(x) - b_2(x))$  (by Lemma 14, (iv)),
- (iii)  $\delta_p(z(p(x))^k + b_1(x) - d(x)) = \delta_p((z - a_2(x))p^k(x) + (b_1(x) - b_2(x))) = \delta_p(b_1(x) - b_2(x))$  (by Lemma 14, (iv)).

(b) if  $b_1(x) = b_2(x)$ , then  $a_1(x) \neq a_2(x)$ , since  $0 \leq \deg a_1(x)$ ,  $\deg a_2(x) < \deg p(x)$ , so

- (i)  $\delta_p(c(x) - d(x)) = \delta_p((a_1(x) - a_2(x))p^k(x)) = \delta_p(a_1(x) - a_2(x)) + k = k$  (by Lemma 14, (i)),
- (ii)  $\delta_p(b_1(x) - d(x)) = \delta_p((-a_2(x))p^k(x)) = \delta_p(a_2(x)) + k = k$  (by Lemma 14, (i)),
- (iii)  $\delta_p(z(p(x))^k + b_1(x) - d(x)) = \delta_p((z - a_2(x))p^k(x) + (b_1(x) - b_2(x))) = \delta_p(z - a_2(x)) + k = k$  (by Lemma 14, (i)).

By the above equations, we obtain

$$\begin{aligned} \delta_p(c(x) - d(x)) &= \delta_p(b_1(x) - d(x)) \\ &= \delta_p(z(p(x))^k + b_1(x) - d(x)). \end{aligned} \quad (102)$$

and by (D2) in Definition 15, we have

$$\delta_p(F(x, c(x))) > \delta_p(F(x, z(p(x))^k + b_1(x))). \quad (103)$$

By (97) and (103), we have

$$\begin{aligned} \delta_p F(x, d(x)) &\geq \delta_p(F(x, c(x))) \\ &> \delta_p(F(x, z(p(x))^k + b_1(x))). \end{aligned} \quad (104)$$

Moreover, we have

$$\begin{aligned} &\delta_p(F(x, z(p(x))^k + b_1(x))) \\ &= \delta_p((z(p(x))^k + b_1(x) - d(x)) \\ &\quad \times F_1(x, z(p(x))^k + b_1(x)) + F(x, d(x))) \\ &\quad \text{(by (98))} \\ &= \delta_p((z(p(x))^k + b_1(x) - d(x)) \\ &\quad \times F_1(x, z(p(x))^k + b_1(x))) \\ &\quad \text{(by (104) and Lemma 14, (iv))} \end{aligned}$$

$$\begin{aligned} &= \delta_p(z(p(x))^k + b_1(x) - d(x)) \\ &\quad + \delta_p(F_1(x, z(p(x))^k + b_1(x))) \quad \text{(by Lemma 14, (i))}, \end{aligned} \quad (105)$$

$$\begin{aligned} &\delta_p(F(x, c(x))) \\ &= \delta_p((c(x) - d(x))F_1(x, c(x)) + F(x, d(x))) \\ &= \delta_p((c(x) - d(x))F_1(x, c(x))) \\ &\quad \text{(by (104) and Lemma 14, (iv))} \\ &= \delta_p(c(x) - d(x)) + \delta_p(F_1(x, c(x))) \\ &\quad \text{(by Lemma 14, (i))}. \end{aligned} \quad (106)$$

From expression (106), we obtain

$$\begin{aligned} &\delta_p(c(x) - d(x)) + \delta_p(F_1(x, c(x))) \\ &= \delta_p(F(x, c(x))) \\ &> \delta_p(F(x, z(p(x))^k + b_1(x))) \quad \text{(by (104))} \\ &= \delta_p(z(p(x))^k + b_1(x) - d(x)) \\ &\quad + \delta_p(F_1(x, z(p(x))^k + b_1(x))) \quad \text{(by (105))}. \end{aligned} \quad (107)$$

From expression (102), that is,  $\delta_p(c(x) - d(x)) = \delta_p(z(p(x))^k + b_1(x) - d(x))$  and by canceling  $\delta_p(c(x) - d(x))$  to both sides of (107), it follows that

$$\delta_p(F_1(x, c(x))) > \delta_p(F_1(x, z(p(x))^k + b_1(x))). \quad (108)$$



That is,  $c(x) \in T_k(F_1)$ . Then we have  $T_k(F) \subseteq T_k(F_1) \cup \{d(x)\}$ . Therefore,

$$|T_k(F)| \leq |T_k(F_1)| + 1 \leq (n - 1) + 1 = n. \tag{109}$$

By induction, the proof is completed.  $\square$

### 5. Applications (As a Game Problem)

A quasi-fixed polynomial problem can be employed to many aspects of actuarial science, risk management, game theory, and so forth. We give an example from game theory as follows.

*Example 1.* Let  $A$  be an insurance company and  $B$  a customer. They make a game as follows.

The companies  $A$  and  $B$  enter into an  $s$ -year contract that within with an interest rate  $x = i$ . Assume each year  $B$  pays a total of  $f_k(i)$  at  $k$ -th year to the insurance company  $A$ ,  $0 \leq k \leq s$  and  $A$  prepare the asset  $S$  for  $B$  at the starting time of this contract. Assume the live inflation rate is  $y = 1 + r$  with  $r = r(i)$  dependent on the interest  $i$ . The total amount that  $B$  pays at the  $s$ -th year of this contract obeys the law of (year) motion about a quasi-fixed polynomial function:

$$F(i, y(i)) = f_s(i)(1+r)^s + f_{s-1}(i)(1+r)^{s-1} + \dots + f_0(i). \tag{110}$$

Now  $x = i, y = y(x)$ .  $p(x) = 1 + x + x^2$  is an irreducible polynomial. When the quasi-fixed polynomial equation

$$F(i, y(i)) = S(1 + i + i^2)^m, \quad m \in \mathbb{N}, \tag{111}$$

then  $B$  can obtain a premium

$$S(1 + i + i^2)^m. \tag{112}$$

The second example concerns actuarial science.

*Example 2.* Traditionally, the disaster occurrence rate of all policies issued by general insurance companies is fixed. The actuary can use a variety of accounting systems to calculate the relationship between the interest rate and disaster occurrence for the policy. But, in general, measurement is very time consuming, and it would be preferable to use a simple mathematical calculation to estimate the relationship between the interest rate and disaster occurrence. In the following, we consider an  $s$ -year pure endowment contract.

An insurance company  $A$  signs a contract to some company  $B$  to the effect that if, in  $k$ -th year, a disaster occurs,  $A$  will pay an amount of money,  $a_k$ , to  $B$ , that  $a_k$  represents the compensation paid by  $A$  to  $B$  at the end of the  $k$ -th year,  $0 \leq k \leq s$  and the disaster occurrence rate is  $q$ . In calculating the insurance fee  $B$  should pay to  $A$  at the beginning of the

contract, we assume that  $A$  gives a bonus interest rate,  $i + i^2$  to  $B$ .

*Question.* How many dollars do  $B$  pay to  $A$  at the beginning of this contract and what is the relationship between the interest rate and disaster occurrence in the policy?

*Answer.*  $A$  pays  $a_k$  dollars if the disaster occurs at the end of  $k$ -th year,  $k = 1, 2, \dots, s$ . Note that

- (1) the probability of the disaster occurring in the  $k$ -th year is  $q(1 - q)^k$ ,
- (2) according to time factor, the value of  $a_k$  dollars at the end of  $k$ -th year is equal to the value of  $a_k(1 + i)^{s-k}$  dollars at the end of  $s$ -th year where  $s \geq k$ , so the amount that  $A$  is expected to pay to  $B$  in the  $k$ -th year is

$$a_k(1 + i)^{s-k}q(1 - q)^k. \tag{113}$$

Then the value of total amount that  $A$  pays  $B$  at the end of  $s$ -th year is

$$F(i, q) = a_0(1 + i)^s q + a_1(1 + i)^{s-1}q(1 - q) + \dots + a_s. \tag{114}$$

This can be rewritten as

$$(i) \quad F(i, q) = f_s(i)q^s + f_{s-1}(i)q^{s-1} + \dots + f_0(i). \tag{115}$$

Moreover, we assume that the total amount that  $B$  pays to  $A$  at the beginning of this contract is  $a$ . Then, by translation of the time factor, the value of this total amount that  $B$  pays to  $A$  by the end of the contract can be said to be

$$(ii) \quad a(1 + i + i^2)^{s+1}. \tag{116}$$

To balance each other, we set the values (i) and (ii) as equal. Thus, the relation equation becomes

$$F(i, q) = a(1 + i + i^2)^{s+1} \quad \text{for some } a \in \mathbb{R}, \tag{117}$$

where  $q$  and  $i$ , respectively, denote the disaster occurrence rate and the interest rate.

We can let  $x = i, y = q, p(x) = 1 + x + x^2$ , and  $m = s + 1$ , then the above problem can be rewritten as

$$F(x, q) = ap^m(x) \quad \text{for some } a \in \mathbb{R}. \tag{118}$$

This example explains a quasi-fixed (polynomial) problem concerning actuarial science.

### References

- [1] A. K. Lenstra, "Factoring multivariate polynomials over algebraic number fields," *SIAM Journal on Computing*, vol. 16, no. 3, pp. 591-598, 1987.
- [2] S. P. Tung, "Near solutions of polynomial equations," *Acta Arithmetica*, vol. 123, no. 2, pp. 163-181, 2006.
- [3] S. P. Tung, "Algorithms for near solutions to polynomial equations," *Journal of Symbolic Computation*, vol. 44, no. 10, pp. 1410-1424, 2009.

- [4] H.-C. Lai and Y.-C. Chen, "A quasi-fixed polynomial problem for a polynomial function," *Journal of Nonlinear and Convex Analysis*, vol. 11, no. 1, pp. 101–114, 2010.
- [5] S. P. Tung, "Approximate solutions of polynomial equations," *Journal of Symbolic Computation*, vol. 33, no. 2, pp. 239–254, 2002.
- [6] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, Cambridge, UK, 2nd edition, 2003.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

