Euclid’s Algorithm, Guass’ Elimination and Buchberger’s Algorithm

Shaohua Zhang

School of mathematics & System Sciences in Shandong University, People’s Republic of China 250100

The key lab of cryptography technology and information security, Ministry of Education, Shandong University, People’s Republic of China 250100

safeprime@163.com

Abstract: As we know, Euclid’s algorithm, Guass’ elimination and Buchberger’s algorithm play important roles in algorithmic number theory, symbolic computation and cryptography even in science and engineering. The aim of this paper is to reveal again the relations of these three algorithms, and, simplify Buchberger’s algorithm without using multivariate division algorithm. We obtain an algorithm for computing the greatest common divisor of several positive integers, which can be regarded as the generalization of Euclid’s algorithm. This enables us to re-find the Guass’ elimination and further simplify Buchberger’s algorithm for computing Gröbner bases of polynomial ideals in modern Computational Algebraic Geometry.

Key words: Euclid’s algorithm, Guass’ elimination, multivariate polynomial, Gröbner bases, Buchberger’s algorithm

Mathematics Subject Classification 2000: 11A, 11T, 11Y, 12Y, 13P, 68W, 94A

1 An algorithm for computing the greatest common divisor of several positive integers

It is well-known that Euclid began his number-theoretical work by introducing his algorithm (See [**1**]: BookⅦ, Propositions 1 and 2).

Proposition 1 (Book Ⅶ): Two unequal numbers being set out, and the less being continually subtracted in turn from the greater, if the number which is left never measures the one before it until a unit is left, the original numbers will be prime to one another.

Proposition 2 (Book Ⅶ): Given two numbers not prime to one another, to find their greatest common measure.

Propositions 1 and 2 in Book Ⅶ of *Elements* are exactly the famous Euclidean algorithm for computing the greatest common divisor of two positive integers. According to Knuth [**2**], "we might call Euclid's method the granddaddy of all algorithms, because it is the oldest nontrivial algorithm that has survived to the present day".

In Book 7, Proposition 3 of his *Elements* [**1**], Euclid further considered how to compute the great common divisors of three positive integers a, b and c. His method is simple and natural. Namely, firstly, compute $(a,b)=d$, secondly, compute $(c,d)=e$, then $(a,b,c)=e$.This method can be readily generalized to the case for computing the greatest common divisor of several positive integers.

In this paper, we try to give another algorithm. Based on Division algorithm, for any positive integer a and b with $a>b$, we may find an integer r such that $(a,b)=(b,r)$ and $b>r$. Hence, once repeating this process, we always can find (a,b). This enlightens us to find firstly the least among several positive integers $a_1, a_2, ..., a_n$ so as to compute their greatest common divisor. Then, we must try to find integers $b_1, b_2, ..., b_m$ with $m<n$ such that $\min\{a_1, a_2, ..., a_n\} \ge \max\{b_1, b_2, ..., b_m\}$ and $(a_1, a_2, ..., a_n)=(b_1, b_2, ..., b_m)$. Once we achieve this goal, then in a finite number of steps, we can find $(a_1, a_2, ..., a_n)$. The following lemma enables us to present our algorithm.

Lemma: Let $a_1, a_2, ..., a_n$ be positive integers with $a_n = \min\{a_1, a_2, ..., a_n\}$. For $1 \le i \le n-1$, Denote $a_i(\mathrm{mod}\, a_n)$ by $R(a_i, a_n)$. Namely, $R(a_i, a_n) = a_i(\mathrm{mod}\, a_n)$. We have the following:

(1) If $R(a_i, a_n) = 0$ for $1 \le i \le n-1$, then $(a_1, a_2, ..., a_n) = a_n$.

(2) Write $\{R(a_i, a_n) \mid R(a_i, a_n) \ne 0,\ for 1 \le i \le n-1\} = \{a_{n+1}, ..., a_{n+r}\}$ with $r \ge 1$.

Then we have $(a_1, a_2, ..., a_n) = (a_n, a_{n+1}, ..., a_{n+r})$.

Proof: Easy.

The algorithm for computing the greatest common divisor of several positive integers:

Algorithm 1: For given positive integers $a_1, a_2, ..., a_n$, this algorithm finds $(a_1, a_2, ..., a_n)$.

1. Compute $\min\{a_1, a_2, ..., a_n\}$. Without loss of generality, let $a_n = \min\{a_1, a_2, ..., a_n\}$.

2. For $1 \le i \le n-1$, one by one compute $R(a_i, a_n)$. If $R(a_i, a_n) = 0$ for any $1 \le i \le n-1$, then output $(a_1, a_2, ..., a_n) = a_n$ and terminate the algorithm. Otherwise, set

$\{a_1, a_2, ..., a_n\} \leftarrow \{a_n, a_{n+1}, ..., a_{n+r}\}$ and go to Step 1, where

$\{a_{n+1}, ..., a_{n+r}\} = \{R(a_i, a_n) \mid R(a_i, a_n) \ne 0,\ for 1 \le i \le n-1\}$.

Remark 1: The advantage of Algorithm 1 is of that we need not do many divisions. However, we must find the least integer in Step 1. As a result, the total running time of our algorithm is approximately the total running time of Euclid’s algorithm for computing the greatest common divisor of several positive integers.

I learned also that Algorithm 1 has been discovered by Blake, Von zur Gathen and Xu [Private Communication]. They further provided an analysis of the algorithm.

2 The Guass' elimination and the simplified Buchberger's algorithm

The aim of this section is to reveal again the relations among Euclid's algorithm, Guass' elimination and Buchberger's algorithm [**3**]. We also try to simplify Buchberger's algorithm without using multivariate division algorithm.

Firstly, we find that, based on the idea of our algorithm in Section 1, one can compute Gröbner bases of the ideal $I =< f_1,..., f_m >$ when $f_1,..., f_m$ are all polynomials of degree 1 over $F[x_1,..., x_n]$. Why? By using the S-polynomial, one can eliminate the leading terms of two polynomials and get the lesser polynomial under given (decreasing) ordering. This is just the innate character of Division algorithm. And our algorithm in Section 1 satisfies exactly this property. Therefore, one could present a simplified algorithm for finding Gröbner bases $g_1,..., g_k$ of the ideal $I =< f_1,..., f_m >$, where $f_1,..., f_m$ are polynomials of degree 1 over $F[x_1,..., x_n]$.

Algorithm 2 (A simplified Buchberger's algorithm for polynomials $f_1,..., f_m$ of degree 1):

1. For polynomials $f_1,..., f_m$ of degree 1 over $F[x_1,..., x_n]$, find a polynomial (without loss of generality, denote by f_m) such that $L(f_i) \geq L(f_m)$ for $1 \leq i \leq m-1$ under the given monomial ordering $x_1 > ... > x_n$. Note that such a polynomial is not always unique.

Remark 2: We say $L(f) \geq L(g)$ with $L(f) = \alpha x_1^{e_1} ... x_n^{e_n}$ and $L(g) = \beta x_1^{d_1} ... x_n^{d_n}$, if there is an integer j with $1 \leq j \leq n$ such that $e_j \geq d_j$ and for any $1 \leq i < j$, $d_i = e_i$, where $\alpha, \beta \in F$. We say $L(d) | L(f)$ with $L(f) = \alpha x_1^{f_1} ... x_n^{f_n}$ and $L(d) = \beta x_1^{d_1} ... x_n^{d_n}$, if for any $1 \leq i \leq n$, $d_i \leq f_i$, where $\alpha, \beta \in F$ and $0 \leq d_i, f_i$ for $1 \leq i \leq n$ are integers.

2. For $1 \leq i \leq m-1$, one by one compute S-polynomial $S(f_i, f_m)$. If $S(f_i, f_m) \in F$ for some

$1 \le i \le m-1$, then output Gröbner bases $x_1,..., x_n, 1$ of the ideal I and terminate the algorithm.

Set $f_i \leftarrow S(f_i, f_m)$ if $L(f_m) | L(f_i)$. Otherwise, set $f_i \leftarrow f_i$. If for any $1 \le i \le m-1$, $(L(f_m), L(f_i)) = 1$, then find another polynomial among $f_1,..., f_{m-1}$, without loss of generality, denote by f_{m-1}, with $L(f_i) \ge L(f_{m-1})$ for $1 \le i \le m-2$. Repeat this process, until find a polynomial whose leading term is not relatively prime with some. Without loss of generality, denote still such a polynomial by f_m, which is the top priority whose leading term is not relatively prime with some by this program. If there is not such a polynomial, then output Gröbner bases $f_1,..., f_m$ of the ideal I. Otherwise, we get a new set $\{R(f_1, f_m),..., R(f_{m-1}, f_m), f_m\}$. And set $\{f_1,..., f_m\} \leftarrow \{R(f_1, f_m),..., R(f_{m-1}, f_m), f_m\}$, go to Step 1.

Remark 3: Note that if $S(f_i, f_m) \in F$, then $I = F[x_1,..., x_n]$ with its Gröbner bases $x_1,..., x_n, 1$. Note that if the leading terms of $f_1,..., f_m$ are coprime each other, then $f_1,..., f_m$ themselves form Gröbner bases of I since they can not offer new S-polynomials. Hence Algorithm 2 is true.

Clearly, Algorithm 2 is essentially Guass' elimination. In order to simplify Buchberger's algorithm, we need a pretreatment algorithm which description is omitted.

Pretreatment algorithm: For any given polynomials $f_1,..., f_m$ over $F[x_1,..., x_n]$ satisfying

$I =< f_1,..., f_m >\ne F[x_1,..., x_n]$, this algorithm finds polynomials $h_1,..., h_k$ over $F[x_1,..., x_n]$ with $k \le m$ such that $L(h_1),..., L(h_k)$ do not divide each other and $I =< f_1,..., f_m > =< h_1,..., h_k >$.

Algorithm 3 (The simplified Buchberger's algorithm): For polynomials $f_1,..., f_m$ of over $F[x_1,..., x_n]$, Algorithm 3 finds Reduced Gröbner basis of $I =< f_1,..., f_m >$. A Gröbner base is reduced if the leading coefficient of each element of the basis is 1 and no monomial in any element of the basis is in the ideal generated by the leading terms of the other elements of the basis. As we know, Reduced Gröbner basis is unique.

1. Find the polynomials $h_1,...,h_k$ over $F[x_1,...,x_n]$ with $k \le m$ such that $L(h_1),...,L(h_k)$ do not divide each other and $I =< f_1,..., f_m > =< h_1,...,h_k >$ by Pretreatment algorithm. With ordering $x_1 > ... > x_n$, sort order $h_1,...,h_k$. Without loss of generality, assume that $h_1 > ... > h_k$.

2. Reduce $h_1 > ... > h_k$ such that $Lc(h_i) = 1$ for $1 \le i \le k$ and each term of h_i is not divisible by $L(h_j)$ for $1 \le i \ne j \le k$. We denote the reduced $h_1 > ... > h_k$ by $h_1 > ... > h_k$ for convenience. Let $A = \{h_1,...,h_k\}$.

3. For any $1 \le i \ne j \le k$, one by one compute S-polynomial $S(h_i, h_j)$ when $(h_i, h_j) > 1$. If for any $1 \le i \ne j \le k$, $(h_i, h_j) = 1$, then terminate the algorithm and output the Gröbner bases $h_1,...,h_k$. Otherwise, regard the new set $\{h_1,...,h_k\} \bigcup \{S(h_1, h_k),..., S(h_{k-1}, h_k)\}$ as $\{ f_1,..., f_m\}$ and go to Step1 and Step2. Thus, we get the new reduced set $B = \{l_1,...,l_r\}$. If $A = B$, then terminate the algorithm and output the Gröbner bases $h_1,...,h_k$ of $I =< f_1,..., f_m >$. Otherwise, set $A \leftarrow B$ and repeat Step3.

Remark 4: By Hilbert's basis theorem which states that every ideal in the ring $F[x_1,..., x_n]$ is finitely generated, Algorithm 3 gives the Gröbner bases in a finite number of steps. Note that the reduced Gröbner basis is unique. Therefore, Algorithm 3 is true.

Based on Algorithms 1, 2 and 3, one will see the relations among Euclid's algorithm, Guass' elimination and Buchberger's algorithm again --- Guass' elimination is the generalization of Euclid's algorithm, and Buchberger's algorithm is the generalization of Guass' elimination.

It is well-known that the problem how to estimate the complexity of Buchberger's algorithm remained a mystery for over thirty years. Although our algorithm 3 can simplify Buchberger's algorithm without using multivariate division algorithm, we do not know how to estimate its complexity yet.

Acknowledgements

I am happy to record my gratitude to my advisor Prof. Wang Xiaoyun who made a lot of helpful suggestion. I wish to thank Center for Advanced Study in Tsinghua University for supplying convenient condition. This work was supported by the National Basic Research Program (973) of China (No. 2007CB807902) and the Natural Science Foundation of Shandong Province (No. Y2008G23).

References

1. Thomas Little Heath, Euclid: The Thirteen Books of the Elements, Cambridge Univ. Press, Cambridge (1926). See also: T L Heath, The Thirteen Books of Euclid's Elements (3 Volumes) New York, (1956).

2. Donald E. Knuth, The Art of Computer Programming, Volume 1-3, 2nd Edition. Addison-Wesley, (1973).

3. Bruno Buchberger, An Algorithm for Finding the Basis Elements of the Residue Class Ring of a Zero Dimensional Polynomial Ideal, Ph.D. dissertation, University of Innsbruck, (1965). English translation by M. Abramson in Journal of Symbolic Computation, 41(2006) 471-511.