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

Shaohua Zhang

School of Mathematics, Shandong University, Jinan, Shandong, 250100, PRC

Abstract: It is known that Euclid's algorithm, Guass' elimination and Buchberger's algorithm play important roles in algorithmic number theory, symbolic computation and cryptography, and 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.

Keywords: Euclid's algorithm, Guass' elimination, multivariate polynomial, Gröbner bases, Buchberger's algorithm

2000 MR Subject Classification: 11A05; 11T71; 11Y16; 11Y40; 13P10; 68W30;

1 Generalization of Euclid's algorithm

It is well-known that Euclid began his number-theoretical work by introducing his algorithm (See [1]: Book 7, Propositions 1 and 2).

Proposition 1 (Book 7): Two unequal numbers being set out, and

[1]E-mail address: shaohuazhang@mail.sdu.edu.cn

[2]This work was partially supported by the National Basic Research Program (973) of China (No. 2007CB807902) and the Natural Science Foundation of Shandong Province (No. Y2008G23).

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 7): Given two numbers not prime to one another, to find their greatest common measure.

Propositions 1 and 2 in Book 7 of *Elements* [1] 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 *Elements* [1], Euclid further considered how to compute the great common divisors of three positive integers a, b, c. His method is simple and natural. Namely, firstly, compute the great common divisors $(a, b) = d$ of a, b, secondly, compute $(d, c) = e$, 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 for computing the greatest common divisor of several positive integers, which can be regarded as the generalization of Euclid's algorithm.

Based on Division algorithm, for any positive integer a, 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 can always find a, b. This enlightens us to firstly find the least among several positive integers $a_1, ..., a_n$ so as to compute their greatest common divisor, then, we try to find integers $b_1, ..., b_m$ with $m < n$ such that $\min\{a_1, ..., a_n\} = \max\{b_1, ..., b_m\}$ and $(a_1, ..., a_n) = (b_1, ..., b_m)$. Once we achieve this goal, then in a finite number of steps, we can find $(a_1, ..., a_n)$. The following lemma enables us to present an algorithm (see Algorithm 1). The proof of this lemma is straitforward and omitted.

Lemma 1: Let $a_1, ..., a_n$ be positive integers with $a_n = \min\{a_1, ..., a_n\}$. Denote $a_i \mod a_n$ by $R(a_i, a_n)$ for $1 \leq i \leq n-1$. Then we have:
(1) If $R(a_i, a_n) = 0$ for any i $(1 \leq i \leq n-1)$, $(a_1, ..., a_n) = a_n$.
(2) When $R(a_i, a_n) \neq 0$ for some i $(1 \leq i \leq n-1)$, we write

$$\{R(a_i, a_n) | R(a_i, a_n) \neq 0, 1 \leq i \leq n-1\} = \{b_1, ...b_m\}.$$

Then we have $(a_1, ..., a_n) = (b_1, ..., b_m)$ and $n - 1 \geq m \geq 1$.

Algorithm 1: This algorithm finds their greatest common divisor of several positive integers.

Input: A set $A = \{a_1, ..., a_n\}$ of positive integers

Output: $(a_1, ..., a_n)$

Step 1: Compute $\min\{a_1, ..., a_n\}$. Set $b \longleftarrow \min\{a_1, ..., a_n\}$.

Step 2: Compute $\{R(a_i, b)|1 \leq i \leq n\}$ and set $B \longleftarrow \{R(a_i, b)|1 \leq i \leq n\}$. If $B = \{0\}$, output $(a_1, ..., a_n) = b$ and terminate the algorithm. Otherwise, set $A \longleftarrow B \setminus \{0\}$ and go to Step 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.

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

2 Guass' elimination and 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 simplify Buchberger's algorithm without using multivariate division algorithm.

Let $f_1, ..., f_m$ be all polynomials of degree 1 over the unique factorization domain $F[x_1, ...x_n]$, where F is a field. By the idea of Algorithm 1, we notice that it is easy to compute Gröbner bases of the ideal $I = \langle f_1, ..., f_m\rangle$. By using the S-polynomial $S(f_i, f_j)$, one can eliminate the leading terms of two polynomials and get the lower polynomial under given monomial ordering. This is just the nature character of Division algorithm. And Algorithm 1 has exactly this property. Therefore, we can further present an algorithm for finding Gröbner bases of the ideal I.

Remark 2: In this paper, we only consider the decreasing ordering: $x_1 > x_2 > ... > x_n$. Thus, the polynomial $f = x_2^2 + x_1 + x_1x_3 + x_2x_3^3$ should be represented by $f = x_1x_3 + x_1 + x_2^2 + x_2x_3^3$. We denote the leading term of $f \in F[x_1, ...x_n]$ by $L(f)$. Under the given ordering $x_1 > x_2 > ... > x_n$, for any two polynomial f, g, $L(f) = \alpha x_1^{e_1}...x_n^{e_n} \geq L(g) = \beta x_1^{d_1}...x_n^{d_n}$ if and only if there is an integer j with $1 \leq j \leq n$ such that $e_j \geq d_j$, moreover for any $1 \leq i < j$, $e_i = d_i$, where $\alpha, \beta \in F$ and e_k, d_k are all non-negative integers

for $1 \leq k \leq n$. In the inequality $L(f) = \alpha x_1^{e_1}...x_n^{e_n} \geq L(g) = \beta x_1^{d_1}...x_n^{d_n}$, we do not need to compare the coefficients α, β. We say $L(f)$ is divisible by $L(g) \neq 0$ if for any $1 \leq k \leq n$, $e_k \geq d_k$. We write $L(g)|L(f)$ if $L(f)$ is divisible by $L(g)$. We call $L(g) \neq 0$ is co-prime to $L(f) \neq 0$ if for any $1 \leq k \leq n$, $e_k d_k = 0$. Write $(L(f), L(g)) = 1$ if $L(f)$ is co-prime to $L(g)$. Especially, $(x_i, x_j) = 1$ if $i \neq j$. The S-polynomial $S(f,g) = \frac{[L(f),L(g)]}{L(f)} f - \frac{[L(f),L(g)]}{L(g)} g$, where $[L(f), L(g)]$ is the least common multiple of $L(f)$ and $L(g)$. Polynomials $g_1, ..., g_k$ over $F[x_1, ...x_n]$ is called a Gröbner bases of the ideal I if the leading term of any polynomial in I is divisible by the leading term of some polynomial in $\{g_1, ..., g_k\}$ and $\langle g_1, ..., g_k \rangle = I$. Particularly, $1, x_1, ..., x_n$ is a Gröbner bases of $F[x_1, ...x_n]$. Note that every ideal I in $F[x_1, ...x_n]$ has a Gröbner basis but its Gröbner bases might be not unique. A Gröbner bases is reduced if the leading coefficient of each element of the basis is 1 and no monomial in any element of the bases is in the ideal generated by the leading terms of the other elements of the basis. As we know, the reduced Gröbner bases is unique. With these notations and definitions, now, we present our algorithms as follows:

Algorithm 2: For given the ideal $I = \langle f_1, ..., f_m \rangle \subsetneq F[x_1, ...x_n]$, where $f_1, ..., f_m$ over $F[x_1, ...x_n]$ is linear, this algorithm gives Gröbner bases $g_1, ...g_k$ of I.

Input: Linear polynomials $f_1, ..., f_m$ and a monomial ordering

Output: Gröbner bases of the ideal $I = \langle f_1, ..., f_m \rangle$ under given ordering

Step 1: If $(L(f_i), L(f_j)) = 1$ for any $1 \leq i \neq j \leq m$, output Gröbner bases $f_1, ..., f_m$ of the ideal $I = \langle f_1, ..., f_m \rangle$.

Step 2: Find $g \in \{f_1, ..., f_m\}$ such that $L(g) = \min\{L(f_1), ..., L(f_m)\}$ and $(L(g), L(r)) > 1$ for some $r \in \{f_1, ..., f_m\}$ with $r \neq g$.

Step 3: Set $f_i \longleftarrow S(f_i, g)$ if $f_i \neq g$, $(L(f_i), L(g)) \neq 1$. Go to Step 1.

Remark 3: Note that if $(L(f_i), L(f_j)) = 1$ for any $1 \leq i \neq j \leq m$, then $f_1, ..., f_m$ form Gröbner bases of I since they can not offer new S-polynomials. In Step 3, we use a key fact that if $(L(f_i), L(g)) \neq 1$, then f_i and g divide each other since they are linear. Therefore, $\{f_1, ..., f_{i-1}, S(f_i, g), f_{i+1}, ...f_m\}$ can replace $\{f_1, ..., f_{i-1}, f_i, f_{i+1}, ...f_m\}$. Notice that $S(f_i, g) \notin F$ since we beforehand assumed $I = \langle f_1, ..., f_m \rangle \subsetneq F[x_1, ...x_n]$. So, Algorithm 2 is true. When $I = \langle f_1, ..., f_m \rangle = F[x_1, ...x_n]$, $f_1, ..., f_m$ is a Gröbner bases of the ideal I.

Remark 4: Clearly, Algorithm 2 is essentially Guass' elimination. By this idea, one can propose a pretreatment algorithm for simplifying Buch-

berger's algorithm. Note that the aforementioned fact in Remark 3 is also useful for this pretreatment algorithm. Here we omit its detailed description.

Pretreatment algorithm: For any given polynomials $f_1, ..., f_m$ over $F[x_1, ...x_n]$ satisfying $I = \langle f_1, ..., f_m \rangle \neq F[x_1, ...x_n]$, this algorithm finds polynomials $h_1, ...h_k$ over $F[x_1, ...x_n]$ with $k \leq m$ such that $L(h_i)$ and $L(h_j)$ do not divide each other for any $1 \leq i \neq j \leq k$ and $I = \langle h_1, ..., h_k \rangle$. Moreover the leading coefficient $Lc(h_i) = 1$ of $L(h_i)$ for $1 \leq i \leq k$ and each term of h_i is not divisible by $L(h_j)$ for $1 \leq i \neq j \leq k$. We call $h_1, ..., h_k$ the reduced polynomials of $f_1, ..., f_m$. Clearly, any two element in $\{h_1, ..., h_k\}$ are comparable. Without loss of generality, we write $h_1 > ... > h_k$.

Algorithm 3 (The simplified Buchberger's algorithm): For polynomials $f_1, ..., f_m$ over $F[x_1, ...x_n]$ satisfying $I = \langle f_1, ..., f_m \rangle \neq F[x_1, ...x_n]$, Algorithm 3 finds the reduced Gröbner bases of $I = \langle f_1, ..., f_m \rangle$.

Input: A set $A = \{f_1, ..., f_m\}$ of polynomials and a monomial ordering
Output: Gröbner bases of the ideal $I = \langle f_1, ..., f_m \rangle$ under given ordering
Step 1: By the pretreatment algorithm, find the reduced polynomials of A: $h_1, ..., h_k$. Set $B \longleftarrow \{h_1, ..., h_k\}$.
Step 2: Set $C \longleftarrow B \cup \{S(h, g) : S(h, g) \neq 0, (L(h), L(g)) \neq 1, h \neq g, \forall g, h \in B\}$. If $\{S(h, g) : S(h, g) \neq 0, (L(h), L(g)) \neq 1, h \neq g, \forall g, h \in B\} = \emptyset$, terminate the algorithm and output the Gröbner bases $h_1, ..., h_k$ of I.
Step 3: By the pretreatment algorithm, find the reduced polynomials of C: $g_1, ..., g_r$. Set $D \longleftarrow \{g_1, ..., g_r\}$. If $D = B$, terminate the algorithm and output the Gröbner bases $h_1, ..., h_k$ of I. Otherwise, $B \longleftarrow D$, go to Step 2.

Remark 5: Since Hilbert's basis theorem states that every ideal in the ring $F[x_1, ...x_n]$ is finitely generated, hence, in a finite number of steps, we must have $D = B$ and Algorithm 3 holds.

A toy example: Compute the reduced Gröbner bases of the ideal $I = \langle f_1 = x^2 + 2xy, f_2 = xy + 2y^2 - 1 \rangle$ under order $x > y$.
1: Note that $L(f_1)$ and $L(f_2)$ do not divide each other, moreover, $Lc(f_1) = Lc(f_2) = 1$, and $L(f_1) > L(f_2)$ under order $x > y$. By the pretreatment algorithm, first, we only need to consider each term of f_1 is not divisible by $L(f_2)$. Since $2xy$ is divisible by xy, hence, we compute $f_3 = f_1 - 2f_2 = x^2 - 4y^2 + 2$. Clearly, $I = \langle f_2, f_3 \rangle$.
2: Compute S-polynomial $S(f_2, f_3) = 2y^2x - x + 4y^3 - 2y$ since $(L(f_2), L(f_3)) \neq 1$. Now, we get a set $A = \{f_2, f_3, S(f_2, f_3)\} = \{x^2 -$

$4y^2+2, 2y^2x-x+4y^3-2y, xy+2y^2-1\}$.

3: Notice that $2y^2x$ is divisible by xy. So, we compute $f_4 = 2yf_2 - S(f_2, f_3) = x$ and get a set $B = \{f_2, f_3, f_4\} = \{x^2-4y^2+2, xy+2y^2-1, x\}$. Clearly, $I = \langle f_2, f_3, f_4\rangle$. Using the pretreatment algorithm, we continue to reduce the set B since $x|L(f_2)$ and $x|L(f_3)$. It shows immediately that $x, 2y^2-1$ is a Gröbner bases of I. Of course, the reduced Gröbner bases of I is $x, y^2-\frac{1}{2}$.

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 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 very thankful to the referees and Professor Jong Hyuk Park for their comments improving the presentation of the paper, and also to my supervisor Professor Xiaoyun Wang for her suggestions. I wish to thank the key lab of cryptography technology and information security in Shandong University and the Institute for Advanced Study in Tsinghua University, for providing me with excellent conditions.

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, 471-511, (2006).