Rathio Mathematica 10
(1996), 23-29

IL VANTAGGIO DI OPERARE IN UN AMBIENTE FINITO

Mauro Zannetti *

1. Introduzione

La fattorizzazione di un polinomio e la determinazione dei suoi fattori lineari, è un argomento che negli ultimi anni ha riscontrato un grande interesse rivolto alla ricerca di algoritmi di risoluzione di bassa complessità computazionale. Nel presente articolo, testo di una lezione tenuta dall'autore presso la sezione provinciale Mathesis di Teramo nel dicembre del 1995 viene trattato il calcolo degli zeri razionali di un polinomio, tenendo conto che ciò equivale alla determinazione dei fattori lineari. La storia del problema della fattorizzazione di un polinomio, è lunga ed interessante. I primi algoritmi, per trovare i fattori lineari di un polinomio con coefficienti interi, furono presentati da Isaac Newton nel 1707, e dall'astronomo Friederich T.V. Schubert nel 1793. Un importante criterio per determinare l'irriducibilità di un polinomio in $Z[x]$ fu dato da F.C. Eisenstein nel 1850. L Kronecker riscoprì il metodo di Schubert nel 1882 e trovò un algoritmo per fattorizzare polinomi con due o più variabili o con coefficienti in estensioni algebriche. Purtroppo quando, circa cento anni più tardi, questi algoritmi furono implementati sul calcolatore, essi risultarono essere altamente inefficienti. Nel 1967 E. Berlekamp, scoprì un ingegnoso algoritmo che permise di fattorizzare sul campo $Z_p[x]$, p numero primo, il cui tempo di calcolo risultò notevolmente minore dei precedenti. La tecnica, interessante di per sè, mette in risalto il vantaggio di trasportare un problema in un ambiente finito, risolverlo e poi tornare nell' ambiente di provenienza. Introduciamo ora il problema della fattorizzazione lineare. Dato un polinomio $A(x)=\sum_{i=0}^{n} a_i x^i$ di grado n con coefficienti di dimensione arbitraria, $a_i \in Z$ per $i=0,1,\ldots,n$,determiniamo , i numeri

* Liceo Scientifico "V. Pollione" Avezzano (AQ).

razionali s/t, $t \neq 0$ tali che $A(s/t)=0$. Il problema è equivalente a trovare tutti i fattori lineari di $A(x)$; infatti, s/t è uno zero razionale, se e solo se $(tx-s)$ divide $A(x)$. Supponiamo che $A(x)$ sia un polinomio monico. Fattorizzare $A(x)$ significa esprimere $A(x)$ nella forma:

$A(x) = p_1(x)^{e_1} \times p_2(x)^{e_2} \times \ldots \times p_r(x)^{e_r}$,

dove $p_1(x), p_2(x), \ldots, p_r(x)$ sono polinomi monici distinti irriducibili. Una tecnica standard per individuare i fattori di molteplicità maggiore di uno è calcolare il Massimo Comun Divisore tra $A(x)$ e la sua derivata $A'(x)$; sia esso uguale a $d(x)$.

Se $d(x)$ è uguale a 1, sappiamo che $A(x)$ è privo di quadrati, infatti: se $A(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_0 = (V(x))^2 \times W(x)$ allora la sua derivata è

$A'(x) = na_n x^{n-1} + (n-1)a_{n-1}x^{n-2} + \ldots + a_1 =$

$V(x)\left[2V'(x)W(x) + V(x)W'(x)\right]$

e $A(x)$ è il prodotto di polinomi primi $p_1(x), p_2(x), \ldots, p_r(x)$. Se $d(x)$ non è uguale a 1, allora $d(x)$ è un fattore proprio di $A(x)$; e così il processo continua fattorizzando $d(x)$ e $A(x)/d(x) = A_1(x)$ separatamente. Si noti che $A_1(x)$ è ora sicuramente privo di quadrati, e su $d(x)$ dovrà essere eseguito lo stesso processo. L'algoritmo di Newton sfrutta il seguente teorema.

Teorema. Sia $A(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_0$ un polinomio in $Z[x]$ e $x = s/t$ una sua radice, con s, t interi relativamente primi, allora s/a_0 e t/a_n.

Dim.: Per ipotesi si ha $A(s/t)=0$ allorra $a_n(s^n/t^n) + a_{n-1}(s^{n-1}/t^{n-1}) + \ldots + a_1(s/t) + a_0 = 0$, moltiplicando tutto per t^n si ha : $a_n s^n = -t\left[a_{n-1}(s^{n-1}) + \ldots + a_1(st^{n-2}) + a_0 t^{n-1}\right]$ per cui t deve dividere $a_n s^n$ e poichè s, t sono relativamente primi, t deve dividere a_n. Ugualmente s deve dividere $a_0 t^n$ e per la stessa ragione s deve dividere a_0.

L'algoritmo è basato sulla ricerca esaustiva tra tutti i fattori di a_0, a_n che potrebbero essere eventuali soluzioni di $A(x)$=0.

Il metodo, ovviamente, è utile solamente per polinomi con a_0 e a_n, interi con poche cifre, per cui ridefiniamo il problema in un dominio immagine con elementi di dimensione fissata. Consideriamo Z_p , gli interi modulo p, dove p è un numero primo.

2. Fattorizzazione lineare su $Z_p[x]$

Trasportiamo $A(x)$ in $Z_p[x]$ considerando $A^*(x) \equiv A(x)(mod\ p)$ con coefficienti in Z_p. Applichiamo ora la ricerca esaustiva ai p elementi di Z_p, se $A^*(p_i) = 0 \quad (p_i \in \{0,1,\ldots,p-1\})$ allora p_i è uno zero di $A^*(x)$. In generale $A^*(x)$ avrà più zeri di $A(x)$ infatti, ad esempio, il polinomio X^4+1 non ha zeri in Z, ma ha uno zero in Z_2 il problema è risalire alle soluzioni in Z partendo dalle soluzioni in Z_p. Le tecniche più utilizzate sono due: quella classica dei Resti Cinesi e quella più moderna del lifting.

Lifting lineare

Teorema. Sia $A(x)$ un polinomio a coefficienti in Z e sia a uno zero semplice di $A(x)(mod\ p)$ con p primo tale che a non è uno zero di $A'(x)(mod\ p)$. Allora $\forall r \geq 1$ l'equazione $A(x) \equiv 0\ mod\ p^r$ ha una soluzione $a_r \equiv a(mod\ p)$.

Dim: procediamo per induzione: per $r = 1$ il teorema è vero per ipotesi. Supponiamo vera l'ipotesi per $r \leq n$. Sia $b_n = A(a_n)/p^n$ la divisione è esatta per l'ipotesi di induzione e sia $a_{n+1} = a_n + p^n y$ con y indeterminata.

Allora
$$A(a_{n+1}) \equiv \left(A(a_n) + p^n y A'(a_n)\right)\left(mod\ p^{n+1}\right) \equiv$$
$$p^n\left(b_n + A'(a)\right)\left(mod\ p^{n+1}\right).$$

Dal momento che per ipotesi si ha $A'(a) \equiv 0\ mod\ p$, possiamo scegliere $y = \left(-b_n / A'(a)\right) mod\ p \qquad y = \left(-b_n / A'(a)\right) mod\ p$ ottenendo $A(a_{n+1}) \equiv 0\ mod\ p^{n+1}$. Per cui il passo di lifting sarà:

$$a_{n+1} = a_n - p^n\left(\left(\left(A(a_n)/p^n\right)/A'(a)\right)(mod\ p)\right).$$

Lifting Quadratico

E' una modifica del lifting lineare. Il teorema che segue, di cui omettiamo la dimostrazione, illustra tale metodo che è più veloce di quello lineare, in quanto sono necessari meno passi per raggiungere uno stesso limite.

Teorema. Sia $A(x)$ un polinomio a coefficienti in Z e a uno zero di $A(x)(mod\ p)$ con p primo tale che a non sia uno zero di $A'(x)(mod\ p)$. Allora $\forall r \geq 1$ l'equazione $A(x) \equiv 0\left(mod\ p^{2^r}\right)$ ha una soluzione $a_{2r} \equiv a(mod\ p)$.

3. Polinomi non monici

Rimuoviamo ora la condizione di monicità sul polinomio. Consideriamo il polinomio $A(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_0$ con coefficienti in Z e scegliamo un primo p tale che p non divide a_n e consideriamo $A_1(x) \equiv A(x)/a_n(mod\ p)$. Sia a_1 uno zero di $A_1(x)$; utiliziamo le tecniche di lifting ricaviamo a_n^* il corrispondente zero di $A_n(x)\left(mod\ p^n\right)$. Il seguente teorema illustra il legame tra a_n^* e la corrispondente soluzione del tipo s/t di $A(x)$.

Teorema. Sia $A(x)$ un polinomio a coefficienti in Z, p un primo tale che p non divide a_n e n un intero positivo. Se $A(x)$ ha uno zero razionale $b = s/t$ allora il polinomio $A_N(x) \equiv A(x)/a_n\left(mod\ p^N\right)$ ha uno zero b_N in Z_{p^N} e $b_N \equiv s\bar{a}_n a_n^{-1}\left(mod\ p^N\right)$, dove $\bar{a}_n$ è l'intero a_n/t.

Dim: Poichè $A(s/t) = 0$ segue che $a_n = \bar{a}_n \times t$. Siccome per ipotesi p non divide a_n, segue che p non divide t e quindi sia a_n che t hanno inversi in Z_{p^N} per ogni N maggiore di zero; pertanto $b_N \equiv s \times t^{-1} \equiv s \times t^{-1} \times a_n \times a_n^{-1} = s \times \bar{a}_n \times a_n^{-1}\left(mod\ p^N\right)$.

poichè l'inverso di questo teorema è falso, dobbiamo sempre verificare che la soluzione s/t, ottenuta da b_N ponendo $s = b_n \times a_n\left(mod\ p^N\right)$ e $t = a_n\left(mod\ p^N\right)$ sia effettivamente uno zero di $A(x)$.

4. Algoritmo lineare di Newton-Hensel-Loos

Sia dato un polinomio privo di zeri multipli $A(x) \in Z[x]$; il seguente algoritmo restituisce una lista L dei suoi zeri razionali;

1) cercare il più piccolo primo p tale che $A(x)(mod\ p)$ è privo di quadrati;
2) sia $b = 2|a_0\|a_n|$ e scegliamo il più piccolo N tale che b sia minore di p^N;
3) calcoliamo la lista L^* degli zeri di $a_n{}^{-1}A(x)(mod\ p)$ mediante l'algoritmo di Newton-Hensel;
4) eleviamo gli zeri in L^* di $a_n{}^{-1}A(x)(mod\ p)$ agli zeri di $a_n{}^{-1}A(x)\left(mod\ p^N\right)$ (lifting lineare);
5) per ogni a^* di L^* riduciamo $\left(a^* a_n\left(mod\ p^N\right)\right)/a_n(mod\ p)$ al termine s/t e se $A(s/t) = 0$, aggiungi s/t a L.

Come miglioramento dell'algoritmo precedente possiamo modificare il passo 4) sostituendo l' algoritmo di lifting lineare con quello quadratico. Gli algoritmi sono stati implementati su sistema Vax / 750, presso il Centro di calcolo dell'Università degli Studi de L'Aquila. L'implementazione è stata effettuata utilizzando il linguaggio Macsyma (Man and Computer Symbolic Manipolation System), linguaggio sviluppato presso il MIT, che è uno dei linguaggi di manipolazione algebrica più potenti tra quelli attualmente disponibili.
Qui di seguito riportiamo le tabelle con i tempi di esecuzione relative all' algoritmo lineare e a quello quadratico.

Algoritmo lineare

grado dei polinomi	*lunghezza delle soluzioni in cifre (tempi di esecuzione)*		
	15	20	40
2	1'	1' 40"	2' .7"
3	1' 2"	2' 1"	5' 6"
5	3' 3"	5' 1"	13' 2"
10	20' 2"	28' 8"	2h 0' 6"

Algoritmo quadratico

grado dei polinomi	*lunghezza delle soluzioni in cifre (tempi di esecuzione)*		
	15	20	40
2	31"	33"	46"
3	43"	47"	1' 3"
5	1' 10"	1' 12"	2' 3"
10	4' 4"	5' 12"	12' 5"
20	19' 1"	26' 13"	53' 3"

Dalle tabelle si può notare come l'algoritmo quadratico sia asintoticamente più efficiente di quello lineare. Ad esempio per polinomi di grado 20 con soluzioni di 40 cifre decimali il tempo di calcolo dell'algoritmo quadratico è circa la metà di quello lineare.

BIBLIOGRAFIA

1 R. BOGEN, *Macsyma Reference Manual,* The Mathlab Group Laboratory For Computers Science M.I.T. Version 1983.

2 J. CALMET, R. LOOS, *Deterministic Versus Probabilistic Factorization of integral Polinomials* 1981.

3 D. S. HIRSCHBERG, C. k. WONG, *A polinomial Time Algorithm for the Knapsack Problem with two variables*, JACM 1976.

4 E. KALTOFEN, NEWARK, DELAWARE, *Factorization of polynomials,* Computing, Suppl. 1982.

5 D. E. KNUTH, *The Art of Computer Programming,* (vol. 2)- Addison Wesley 1969.

6 R. LOOS, *Computing Rationals Zeros of integral Polinomials by p-adic Expantion*-SIAM J. Computing 1983.

7 R. PAVELLE, M. ROTHSTEIN, J.FITCH, *Computer Algebra,* Scientific American 1981.

8 R. H. RAND, *Computer Algebra in Applied Mathematics,* An introduction to Macsyma-Research Notes in Mathematics n. 94, Pitman Advanced Publishing Program 1984.

9 D. Y. Y. YUNG, *Algebraic Algorithms using p-adic Construction,* ACM Symposium 1976.