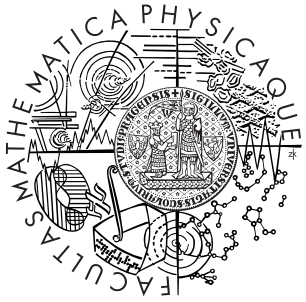
Univerzita Karlova v Prahe
Matematicko-fyzikálna fakulta

BAKALÁRSKA PRÁCA



Ján Jančo

Nastavenie konštant pri hľadaní polynómov pre Číselne teoretické sito

Katedra algebry

Vedúci bakalárskej práce: RNDr. Přemysl Jedlička, Ph.D., Katedra matematiky, Technická fakulta, Česká zemědělská univerzita

Študijný program: Matematika, Obecná matematika

2009

Na tomto mieste by som sa chcel poďakovať vedúcemu mojej bakalárskej práce RNDr. Přemyslovi Jedličkovi za cenné rady a pripomienky. Ďalej by som sa chcel poďakovať tvorcom softvérovej implementácie algoritmu NFS, s ktorou som pracoval. V neposlednom rade by som chcel vyjadriť vďaku Centru Jindřicha Nečase pro matematické modelování za možnosť vykonávať všetky počítačové výpočty na Karlínskom clusteri Sněhurka.

Prehlasujem, že som svoju bakalársku prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce a jej zverejňovaním.

V Prahe dňa 6. 8. 2009 Ján Jančo

Obsah

Názov práce: Nastavenie konštant pri hľadaní polynómov pre Číselne teoretické sito
Autor: Ján Jančo
Katedra (ústav): Katedra algebry
Vedúci bakalárskej práce: RNDr. Přemysl Jedlička, Ph.D.
e-mail vedúceho: jedlickap@tf.czu.cz

Abstrakt: V tejto práci sa zaoberáme algoritmom Číselne teoretické sito. Sústredíme sa hlavne na jeho prvú fázu, Hľadanie polynómov, ktorej úlohou je nájsť čo najvhodnejšie polynómy potrebné v ďalšej časti algoritmu. Rozoberieme aké polynómy a prečo sú vhodné a popíšeme jeden z algoritmov na ich hľadanie. Výsledok tohto algoritmu podstatne závisí od počiatočných nastavení jeho parametrov. Cieľom práce je pochopiť, ako ovplyvňujú rôzne nastavenia týchto parametrov výsledok algoritmu a zároveň určiť ich optimálne nastavenie.

Kľúčové slová: Číselne teoretické sito, Hľadanie polynómov, faktorizácia

Title: Constants setting for the Number Field Sieve polynomial selection
Author: Ján Jančo
Department: Department of algebra
Supervisor: RNDr. Přemysl Jedlička, Ph.D.
Supervisor's e-mail address: jedlickap@tf.czu.cz

Abstract: In this work we study the Number Field Sieve algorithm for large number factorization. We concentrate mainly on its first stage, Polynomial Selection. In this phase the algorithm tries to find the most suitable polynomials needed for the next step. Further we will discuss which polynomials and why suit the best and describe a method for finding them. The results and the running times are subtantially dependent on the initial settings of the algorithm's parameters. The aim of this work is to understand how the results given by the algorithm are influenced by these settings. We use this knowledge to get the optimal ones.

Keywords: Number Field Sieve, Polynomial Selection, factorization

Kapitola 1

Úvod

Problém faktorizácie čísla, t.j. jeho rozkladu na súčin prvočísel, je starý, napriek tomu dodnes uspokojivo nevyriešený, matematický problém. Jeho vyriešenie by okrem významného pokroku na poli matematickej teórie spôsobilo obrovský prielom v oblasti šifrovanej komunikácie. Bezpečnosť RSA, v súčasnosti najpoužívanejšej asymetrickej šifry, je totiž založená na nemožnosti faktorizovať príliš veľké číslo (rádovo niekoľko sto ciferné) v prijateľnom čase.

Nájsť prvočíselný rozklad nie je rovnako obtiažny problém pre každé číslo. Pre niektoré čísla to môže byť jednoduchšia úloha, pre iné naopak omnoho komplikovanejšia. Faktorizačné algoritmy sa preto delia do dvoch hlavných kategórii. Do prvej spadajú tie, ktoré sú vhodné pre čísla špeciálneho tvaru. Doba ich výpočtu väčšinou významne závisí na veľkosti deliteľa daného čísla. Do druhej kategórie patria algoritmy určené na faktorizáciu ľubovoľného čísla. Ich doba výpočtu závisí len na veľkosti faktorizovaného čísla.

Významný pokrok v algoritmoch z druhej kategórie nastal v 2. polovici 20. storočia. Najskôr sa v roku 1975 objavil algoritmus CFRAC (*Continued fraction method*) vyvinutý matematikmi M. A. Morrisonom a J. Brillhartom.[1] Ten bol schopný v prijateľnom čase faktorizovať 50-ciferné čísla. V roku 1981 predstavil C. Pomerance algoritmus *Kvadratické sito*[2], ktorý po istých vylepšeniach dokáže faktorizovať viac ako 100-ciferné čísla. Nakoniec v roku 1988 J. Pollard vyvinul *Special number field sieve*, ktoré bolo neskôr zovšeobecnené na Číselne teoretické sito[3] (NFS), v súčasnosti najrýchlejší faktorizačný algoritmus pre čísla rádu (počet cifier) 130 a viac.

Pre ilustráciu veľkosti pokroku, v roku 1977 R. Rivest uverejnil odhad, podľa

[1] Jeho myšlienka bola predstavená už v roku 1931 D. H. Lehmerom a R. E. Powersom.

[2] z angl. *Quadratic sieve*

[3] z angl. *Number field sieve* niekedy označovaného ako *General number field sieve*

ktorého by faktorizácia 125-ciferného čísla na počítači trvala $40 \cdot 10^{15}$ rokov. To všetko za predpokladu, že by počítač vykonával modulárne násobenie behom jednej nanosekundy [8]. Už v 1994 sa ale podarilo pomocou Kvadratického sita rozložiť na súčin prvočísel 129-ciferné číslo.[4] V súčasnosti je držiteľom faktorizačného rekordu algoritmus NFS, pomocou ktorého bol v 2005 nájdený prvočíselný rozklad 200-ciferného čísla [9].

Táto práca sa zaoberá práve Číselne teoretickým sitom. V Kapitole 2 popíšeme hlavnú myšlienku tohto algoritmu. V 3. kapitole sa budeme podrobnejšie venovať jeho prvej fáze, hľadaniu vhodných polynómov a jej vplyvu na ďalší priebeh algoritmu. Na záver v Kapitole 4 pomocou experimentov odvodíme pre čísla v určitom rozsahu optimálne nastavenie konštant ovplyvňujúcich výsledok prvej fázy.

Na záver sa krátko venujme terminológii pojmu Number Field Sieve, ktorú budeme používať v tejto práci. Slovenský ekvivalent pre NFS je *Sito číselných polí* alebo *Všeobecné sito číselných polí* (General Number Field Sieve, GNFS). Tieto preklady nie sú ale na Slovensku plne udomácnené a zvyknú sa používať aj pôvodné anglické názvy alebo skratky (NFS, resp. GNFS). Český ekvivalent pre NFS je *Číselně teoretické síto*. V tejto práci budeme používať slovenský preklad českého pojmu, t.j. *Číselne teoretické sito* alebo anglickú skratku NFS.

[4] Celý výpočet trval viac ako 6 mesiacov na 1600 počítačoch.

Kapitola 2

Algoritmus Číselne teoretické sito

2.1 Základná myšlienka

Už v 17. storočí francúzsky matematik Pierre de Fermat navrhol stratégiu, akou by bolo možné faktorizovať číslo N. Jeho myšlienka bola jednoduchá. Ak nájdeme dve prirodzené čísla x, y také, že

$$N = x^2 - y^2, \tag{2.1}$$

potom $x + y$ a $x - y$ sú delitele N. Aby sme sa vyhli triviálnemu výsledku, t.j. $x + y = N$ a $x - y = 1$, musí byť $x \neq (N + 1)/2$ a $y \neq (N - 1)/2$. Takáto dvojica x, y existuje pre každé N, ktoré sa dá napísať ako súčin dvoch párnych alebo dvoch nepárnych čísel. Ak totiž $N = pq$ (p, q sú buď obidve párne alebo obidve nepárne), potom

$$\left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2 = \left(\frac{p+q}{2} + \frac{p-q}{2}\right)\left(\frac{p+q}{2} - \frac{p-q}{2}\right) = N.$$

Tým Fermat previedol úlohu faktorizácie čísla N na úlohu hľadania reprezentácie N ako rozdielu druhých mocnín dvoch prirodzených čísel. Pre N veľké je ale aj táto úloha veľmi obtiažnym problémom.

S vylepšením Fermatovej myšlienky prišiel Kraitchik v prvej polovici 20. storočia. Oslabil podmienku (2.1) tým, že od x, y požadoval, aby spĺňali „iba“ kongruenciu tvaru

$$x^2 \equiv y^2 \pmod{N}. \tag{2.2}$$

Kongruencii (2.2) sa niekedy hovorí *Legendreova kongruencia*. Ak pre x, y spĺňajúce (2.2) naviac platí $1 < \mathrm{NSD}(x + y, N) < N$ alebo $1 < \mathrm{NSD}(x - y, N) < N$, našli

sme netriviálneho deliteľa čísla N. Nie je ťažké ukázať, že takáto situácia nastáva aspoň pre polovicu zo všetkých riešení (2.2).

Zatiaľčo množina riešení rovnice (2.1) je konečná, Legendreova kongruencia má nekonečne veľa riešení, teda aj nekonečne veľa riešení dávajúcich netriviálny rozklad čísla N. To nám dáva šancu, že hľadanie takýchto dvojíc bude ľahšie.

Na princípe hľadania dvojíc (x, y) spĺňajúcich (2.2) sú založené mnohé moderné faktorizačné metódy. Spomedzi nich uveďme napríklad už v úvode spomínané CFRAC, Kvadratické sito a Číselne teoretické sito. Spomenuté algoritmy sa od seba líšia metódou, akou sa hľadajú riešenia kongruencie (2.2). Ich stručný popis môžeme nájsť napríklad v [7]. Vo zvyšku kapitoly sa budeme venovať len NFS a vysvetlíme princíp, ako tento algoritmus funguje. Pre hlbšie štúdium tohto algoritmu odporúčame [4].

2.2 Algoritmus NFS

Číselne teoretické sito patrí v súčasnosti medzi najkomplikovanejšie faktorizačné algoritmy. Využíva netriviálnu algebraickú teóriu čísel. Skôr než sa ale pustíme do popisu tohto algoritmu, ukážeme si jednoduchý princíp ako skonštruovať druhú mocninu nejakého prirodzeného čísla z tzv. *hladkých prirodzených čísel*. Tento postup neskôr využijeme v NFS.

2.2.1 Konštrukcia druhej mocniny prirodzeného čísla z hladkých prirodzených čísel

Definícia 2.1. *Nech n, B sú prirodzené čísla. Povieme, že n je B-hladké, ak každé prvočíslo p, ktoré delí n, je menšie ako B. Číslo B sa nazýva hranica hladkosti. Ak hodnota B je abstraktná, hovoríme o hladkých prirodzených číslach.*

Predpokladajme, že máme danú množinu B-hladkých čísel, označme ich $n_i, i = 1, \ldots, m$, kde m je počet prvkov danej množiny. Nech ďalej $m > \pi(B)$ (počet prvočísel menších ako B). Potom

$$n_i = p_1^{e_{i1}} \cdot \ldots \cdot p_{\pi(B)}^{e_{i\pi(B)}}, \text{ pre vhodné } e_{ij} \geq 0, j = 1, \ldots, \pi(B),$$

kde $p_1, \ldots, p_{\pi(B)}$ sú všetky prvočísla menšie ako B. Našou úlohou je teraz vybrať také čísla n_{i_k}, ktorých súčin bude druhou mocninou nejakého prirodzeného čísla. Inak povedané, chceme nájsť binárny vektor $(a_1, \ldots, a_m)$ taký, že

$$\prod_{i=1}^{m} n_i^{a_i} = p_1^{\sum_{i=1}^{m} a_i e_{i1}} \cdot \ldots \cdot p_{\pi(B)}^{\sum_{i=1}^{m} a_i e_{i\pi(B)}} = l^2 \text{ pre nejaké } l \in \mathbb{N}.$$

To nastáva práve vtedy, keď sú všetky exponenty daných prvočísel p_i párne. Tým sa problém zredukoval na hľadanie riešenia sústavy lineárnych rovníc nad $\mathbb{Z}_2$

$$\begin{aligned} a_1\, e_{11} + a_2\, e_{21} + \ldots + a_m\, e_{m1} &= 0 \\ &\vdots \\ a_1\, e_{1\pi(B)} + a_2\, e_{2\pi(B)} + \ldots + a_m\, e_{m\pi(B)} &= 0. \end{aligned} \tag{2.3}$$

V maticovom zápise dostávame

$$\left(\begin{array}{cccc|c} e_{11} \bmod 2 & e_{21} \bmod 2 & \ldots & e_{m1} \bmod 2 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ e_{1\pi(B)} \bmod 2 & e_{2\pi(B)} \bmod 2 & \ldots & e_{m\pi(B)} \bmod 2 & 0 \end{array}\right).$$

Na nájdenie riešenia sústavy (2.3) použijeme vhodnú metódu. Predpoklad $m > \pi(B)$ zaručuje existenciu netriviálneho riešenia, teda aj netriviálneho výberu čísel n_{i_k}, ktorých súčin je druhou mocninou nejakého prirodzeného čísla.

V nasledujúcom oddiele uvidíme, k čomu je vhodná takáto konštrukcia v NFS. Na záver ešte dodajme, že pri aplikácii v algoritme NFS je spomínaná matica riedka a obrovských rozmerov. Preto sa pri riešení danej úlohy využíva blokový Wiedermannov alebo blokový Lanczosov algoritmus. Pomocou nich nezískame bázu priestoru všetkých riešení, ale iba určitý počet vektorov z priestoru všetkých riešení.

2.2.2 Hlavná myšlienka NFS

Nech $f_1(x), f_2(x) \in \mathbb{Z}[x]$ sú ireducibilné polynómy nad $\mathbb{Q}$, ktoré majú spoločný koreň $m \in \mathbb{Z}$ modulo N, t.j.

$$f_1(m) \equiv f_2(m) \equiv 0 \pmod{N}.$$

Nech ďalej θ_i je komplexný koreň polynómu f_i, $i = 1, 2$. Definujme polynómy

$$F_i(a, b) := b^{\deg f_i} f_i\left(a/b\right),\ i = 1, 2$$

a okruhové homomorfizmy $\varphi_i : \mathbb{Z}[\theta_i] \to \mathbb{Z}_N$ predpisom

$$\varphi_i : a + b\theta_i \mapsto a + bm \bmod N,\ \text{pre } a, b \in \mathbb{Z},\ i = 1, 2.$$

Predpokladajme, že K je množina dvojíc (a, b), kde $a, b \in \mathbb{Z}$ sú nesúdeliteľné a platí

$$\prod_{(a,b)\in K} (a - b\theta_i) = \beta_i^2 \text{ pre nejaké } \beta_i \in \mathbb{Z}[\theta_i], i = 1, 2.$$

Potom

$$\varphi_i(\beta_i)^2 = \varphi_i\left(\beta_i^2\right) = \prod_{(a,b)\in K} \varphi_i\left(a - b\theta_i\right) \equiv \prod_{(a,b)\in K} (a - bm) \pmod{N},$$

pre $i = 1, 2$, z čoho dostávame $\varphi_1(\beta_1)^2 \equiv \varphi_2(\beta_2)^2 \pmod{N}$. Otázkou ostáva, ako nájsť množinu K.

Dá sa dokázať (pre pevne zvolené i), ak je

$$\prod_{(a,b)\in M} F_i(a, b)$$

druhou mocninou prirodzeného čísla pre M množinu nesúdeliteľných dvojíc (a, b), potom s vysokou pravdepodobnosťou je aj

$$\prod_{(a,b)\in M} (a - b\theta_i) = \gamma_i^2 \text{ pre nejaké } \gamma_i \in \mathbb{Q}(\theta_i) \tag{2.4}$$

Dôkaz viď napr. v [5]. Poznamenajme, že práve uvedené tvrdenie je v skutočnosti omnoho komplikovanejšie a my sme vyslovili len jeho hlavnú myšlienku, za ktorou je skrytá komplikovaná algebraická teória čísel.

Množinu K budeme teda konštruovať z hladkých hodnôt polynómov F_i. Ak nájdeme dostatočný počet nesúdeliteľných dvojíc (a, b), pre ktoré sú $F_1(a, b)$ a $F_2(a, b)$ B-hladké (pre predom zvolené B), potom pomocou jednoduchého rozšírenia algoritmu z Oddielu 2.2.1 zostrojíme K tak, aby $\prod_{(a,b)\in K} F_i(a, b)$ bol druhou mocninou prirodzeného čísla pre $i = 1, 2$. Nech teraz platí (2.4) pre $i = 1, 2$ a zároveň pre nejaké $c \in \mathbb{Z}$ je $c\gamma_1 \in \mathbb{Z}[\theta_1]$ a $c\gamma_2 \in \mathbb{Z}[\theta_2]$. Nie je ťažké si rozmyslieť, že potom už

$$\varphi_1(c\gamma_1)^2 \equiv \varphi_2(c\gamma_2)^2 \pmod{N}.$$

Pokiaľ pre nájdené K neplatí (2.4), použijeme inú vhodnú množinu K'. Tú zostrojíme z ďalšieho riešenia sústavy (2.3) (viď Oddiel 2.2.1). Postup opakujeme, kým nedosiahneme platnosť (2.4). Pravdepodobnosť nájdenia vhodnej množiny K sa s rastúcim počtom opakovaní vyššie načrtnutého postupu rýchlo blíži k 1.

Ukázali sme, ako konštruovať riešenia x, y Legendreovej kongruencie. Ak nájdené x, y nám dajú triviálny rozklad N, opäť sa pokúsime nájsť tým istým spôsobom inú množinu K spĺňajúcu (2.4). Aj tu platí, že pravdepodobnosť nájdenia vhodnej dvojice x, y sa rýchlo blíži k 1 s rastúcim počtom opakovaní postupu.

2.2.3 Algoritmus

V predchádzajúcom oddiele sme predviedli, ako NFS konštruuje riešenia Legendreovej kongruencie. Na základe toho môžeme algoritmus NFS rozdeliť na 4 hlavné časti:

1. **Hľadanie polynómov (*Polynomial Selection*):** Nájdenie vhodných ireducibilných polynómov $f_1(x), f_2(x) \in \mathbb{Z}[x]$, ktoré majú spoločný koreň modulo N.

2. **Presievanie (*Sieving*):** Hľadanie nesúdeliteľných dvojíc $a, b \in \mathbb{Z}$, pre ktoré sú $F_1(a, b)$ a $F_2(a, b)$ B-hladké (pre vhodne zvolenú hranicu B).[1]

3. **Redukcia matice (*Matrix Reduction*):** Zostrojenie matice podobne ako v Oddiele 2.2.1, nájdenie viacerých riešení príslušnej homogénnej sústavy lineárnych rovníc a zostrojenie požadovanej množiny K.

4. **Odmocňovanie (*Square Root*):** Pre dané γ_i^2 nájsť $\gamma_i \in \mathbb{Q}[\theta_i]$.

Poznamenajme ešte, že vo fáze Presievanie prebieha hľadanie dvojíc (a, b) len cez vhodne zvolenú $S \subset \mathbb{Z} \times \mathbb{N}$. Množine S sa hovorí *presievacia oblasť*.

Časovo najnáročnejšia je 2. časť. Pre čísla vysokého rádu potrebujeme totiž „nazbierať“ až niekoľko miliónov vhodných dvojíc (a, b). Z toho vyplýva, že aj matica, s ktorou sa pracuje v 3. časti, môže mať milióny riadkov a stĺpcov a nebude jednoduché nájsť požadované riešenia. Ani 4. časť nie je triviálna. Napriek tomu sú pre tieto časti vyvinuté veľmi dobré algoritmy. Výrazné zrýchlenie NFS môžeme docieliť vhodnou voľbou polynómov v 1. fáze. Potrebujeme, aby polynómy F_i „často“ nadobúdali hladké hodnoty. Tým by sme mohli výrazne zrýchliť 2. časť a teda aj celý algoritmus.

Voľba polynómov je v súčasnosti hlavný otvorený problém algoritmu NFS. Nie je totiž problém nájsť polynómy f_1, f_2 popísané v Oddiele 2.2.2, tých je veľa. Problém je určiť, ktoré z nich sú pre nás „dobré“, vedieť ako ich konštruovať a potom vybrať z nich tie „najlepšie“. Tejto fáze sa budeme podrobnejšie venovať v nasledujúcej kapitole. Definujeme, čo je *dobrý* polynóm, a popíšeme algoritmus na konštruovanie *dobrých* polynómov.

Zložitosť algoritmu

Venujme sa na záver tohto oddielu a celej kapitoly stručne otázke zložitosti algoritmu NFS. Ako sme už spomenuli vyššie, dominantnou časťou NFS je presievanie.

[1] Pre ilustráciu, napr. pre čísla rádu 100 je $B \approx 10^6$ a pre čísla rádu 140 je $B \approx 10^8$.

Preto sa pri úvahách o zložitosti väčšinou analyzuje iba náročnosť fázy Presievanie. Tá sa optimalizuje vhodnou voľbou stupňov polynómov f_1, f_2, ďalej vhodnou voľbou B a presievacej oblasti.

Pre účely odhadu asymptotickej zložitosti presievania sa definuje funkcia

$$L[x; v, w] := \exp\left[(w + o(1))(\log x)^v (\log\log x)^{1-v}\right], v, w \in (0, 1).$$

Optimálnou voľbou vyššie spomínaných parametrov sa docieli asymptotická zložitosť presievania (a teda aj celého algoritmu NFS) rovná $L[N; 1/3, (64/9)^{1/3}]$ (pozri [2] alebo [5]). Teda Číselne teoretické sito má subexponenciálnu zložitosť v závislosti na dĺžke faktorizovaného čísla N.

Zatiaľčo pri analýze asymptotickej zložitosti NFS sa funkcia L objavuje s parametrom $v = 1/3$, ostatné v súčasnosti známe algoritmy dosahujú prinajlepšom $v = 1/2$ (napr. Kvadratické sito). Preto Číselne teoretické sito v súčasnosti poráža všetky ostatné faktorizačné algoritmy.

Kapitola 3

Generovanie polynómov v Číselne teoretickom site

V tejto kapitole sa budeme podrobnejšie venovať prvej fáze algoritmu NFS a to generovaniu polynómov f_1, f_2. Pripomeňme, že polynómy f_1, f_2 sú celočíselné, ireducibilné nad $\mathbb{Q}$ a majú spoločný koreň modulo N, kde N je číslo, ktoré chceme faktorizovať. V Oddiele 3.1 stručne popíšeme jeden z algoritmov na ich konštrukciu, tzv. *base-m metódu*.

Pre jednoduchosť vyjadrovania budeme dvojicu vyššie spomenutých polynómov (f_1, f_2) niekedy označovať ako *polynomiálny pár* a pod pojmom *stupeň polynomiálneho páru* budeme rozumieť dvojicu $(deg\, f_1, deg\, f_2)$.

Z analýzy zložitosti Číselne teoretického sita vyplynula pre polynomiálny pár (f_1, f_2) podmienka na $(\deg f_1, \deg f_2)$ (viď Oddiel 2.2.3). Množina takýchto polynomiálnych párov je ale stále dosť široká. Ľubovoľný výber z tejto množiny (aj ten najlepší) nemá síce významný vplyv na asymptotickú zložitosť NFS, v praxi však môže podstatne ovplyvniť dobu trvania presievacej fázy a teda aj dobu celého výpočtu (faktorizácie čísla N). Inak povedané, ak by sme pre presievanie zvolili „nadpriemerné“ polynómy, výrazne tým urýchlime faktorizáciu N. V Oddiele 3.2 sa preto budeme venovať vlastnostiam polynómov, ktoré sú pre NFS podstatné a popíšeme metódu kvantifikácie ich „kvality“. Nakoniec v Oddiele 3.3 popíšeme algoritmus na konštrukciu nadpriemerných polynómov.

3.1 Konštrukcia polynomiálnych párov

Existuje viacero algoritmov na konštrukciu polynomiálnych párov (f_1, f_2). Líšia sa od seba najmä tým, akého stupňa polynomiálne páry vytvárajú. Jedným z týchto

algoritmov je base-m metóda, ktorej výsledkom je polynomiálny pár stupňa $(d, 1)$ pre ľubovoľné predom zvolené d.

Base-m metóda

Nech $d \geq 1$ je pevné. Zvoľme $m \approx N^{1/(d+1)}$ a označme $a_i^{(m)}$, $i = 0, \ldots, d$ koeficienty reprezentácie N v sústave so základom m, t.j.

$$N = \sum_{i=0}^{d} a_i^{(m)} m^i \,, \qquad 0 \leq a_i^{(m)} < m.$$

Polynómy f_1, f_2 sa definujú nasledovne

$$f_1(x) := \sum_{i=0}^{d} a_i^{(m)} x^i \,, \qquad f_2(x) := x - m.$$

Koeficienty $a_i^{(m)}$, $i = 0, \ldots, d-1$, môžeme ešte zmenšiť nasledujúcou transformáciou. Ak je $a_i > \lfloor m/2 \rfloor$, potom

$$a_i \mapsto a_i - m, \qquad a_{i+1} \mapsto a_{i+1} + 1.$$

Polynómu f_1 sa niekedy hovorí *base-m reprezentácia čísla* N.

Z konštrukcie vyplýva, že $m = O(N^{1/(d+1)})$ a tiež $a_i^{(m)} = O(N^{1/(d+1)})$. Poznamenajme ešte, že namiesto reprezentácie čísla N môžeme voliť reprezentáciu kN pre nejaké malé $k \in \mathbb{Z}$.

Ďalšou technikou na konštrukciu (f_1, f_2) je *Montgomeryho dva-kvadratická metóda*, ktorá produkuje polynomiálny pár stupňa $(2, 2)$. Táto metóda sa dá zovšeobecniť aj na stupne (d, d) pre $d \geq 3$, tu už ale vznikajú problémy, ktoré sťažujú použitie tohto zovšeobecnenia v praxi. Vo všeobecnosti je konštrukcia polynomiálnych párov otvorený problém. Nie je známy algoritmus, ktorý by produkoval kvalitné polynomiálne páry ľubovolného stupňa.

3.2 Kvantifikácia kvality polynómov

Z hľadiska NFS sú polynómy F_1, F_2 tým lepšie, čím viac hladkých hodnôt nadobúdajú v presievacej oblasti. Pripomeňme, že F_1, F_2 sú definované nasledovne

$$F_i(x, y) := y^{\deg f_i} f_i(x/y),\ i = 1, 2.$$

Na pojem *kvalita polynómu* sa teda môžeme pozerať ako na počet B-hladkých hodnôt, ktoré polynóm nadobúda na danej množine pre dané B. Otázkou ostáva, čo ovplyvňuje túto kvalitu.

Vlastnosti, ktoré významne ovplyvňujú výskyt B-hladkých hodnôt a tým aj kvalitu polynómu $F(x, y)$:

- veľkosť hodnôt $|F(x, y)|$ nadobúdaných na presievacej oblasti; intuitívne je jasné, že čím je $r \in \mathbb{N}$ väčšie, tým sa jeho šanca byť B-hladkým zmenšuje
- počet koreňov $F(x, y)$ modulo p pre prvočísla p, tzv. *koreňové vlastnosti*; v nasledujúcich oddieloch ukážeme, ak F má veľa koreňov modulo malé p, pravdepodobnosť, že F nadobúda B-hladké hodnoty sa zvýši.

Podrobnejšie sa budeme venovať týmto vlastnostiam a kvantifikácii ich vplyvu na kvalitu polynómu v oddieloch 3.2.2 a 3.2.3. V Oddiele 3.2.1 stručne rozoberieme *Dickmanovu funkciu*, ktorá slúži k odhadu pravdepodobnosti, že je náhodné číslo B-hladké.

3.2.1 Dickmanova funkcia

Pre $r, B \in \mathbb{N}$ položme

$$\psi(r, B) = |\{n \in \mathbb{N} \mid n \leq r,\ n \text{ je } B\text{-hladké}\}|.$$

Definujme pre $u \in \mathbb{R}$, $u \geq 0$

$$\rho(u) = \lim_{r \to \infty} \frac{\psi(r, r^{1/u})}{r}, u > 1,$$

$\rho(u) = 1$ pre $0 \leq u \leq 1$. Funkcia ρ sa nazýva *Dickmanova funkcia*. Pre ďalšie účely sa nám hodí ešte nasledujúca definícia.

Definícia 3.1. *Náhodná hodnota i_r je prirodzené číslo vybrané uniformne náhodne z množiny $\{i \in \mathbb{N} \mid 1 \leq i \leq r\}$.*

Výraz $\psi(v, v^{1/u})/v$ je pravdepodobnosť, že náhodná hodnota i_v je $v^{1/u}$-hladká. Hodnota $\rho(u)$ je teda asymptotická pravdepodobnosť, že najväčší prvočíselný deliteľ i_v je nanajvýš $v^{1/u}$. Alebo $\rho(u)$ je asymptotická pravdepodobnosť, že i_v je B-hladké pre $u = (\log v)/\log B$.

Vlastnosti Dickmanovej funkcie sú veľmi dobre preskúmané [6]. Nám bude stačiť platnosť nasledujúceho vzťahu

$$\psi(r, r^{1/u}) = r\rho(u) + \frac{r(1-\gamma)\rho(u-1)}{\log r} + O\left(\frac{r}{\log^2 r}\right),$$

kde γ je Eulerova konštanta (viď [3]). Ak $P(r, B)$ označuje pravdepodobnosť, že náhodná hodnota i_r je B-hladká, potom z predchádzajúceho vzťahu dostávame

$$P(r, B) \approx \rho(u) + \frac{(1-\gamma)\rho(u-1)}{\log r}, \quad \text{pre } u = \frac{\log r}{\log B}. \tag{3.1}$$

Na vyčíslenie Dickmanovej funkcie existuje efektívna metóda [1]. To nám umožňuje používať vzťah (3.1) v praxi.

3.2.2 Vplyv veľkosti polynomiálnych hodnôt na kvalitu

Veľkosť hodnôt polynómov F_1, F_2 sa berie do úvahy už pri optimalizácii zložitosti NFS. Intuitívne je jasné, a zo vzťahu (3.1) aj vyplýva, že pravdepodobnosť i_r byť B-hladkou s rastúcim r klesá. Preto potrebujeme zaistiť, aby hodnoty polynómov F_1, F_2 boli na presievacej oblasti čo najmenšie.

V tomto oddiele sa obmedzíme iba na polynómy získané base-m metódou. Teda budeme uvažovať polynomiálne páry (f_1, f_2) stupňa $(d, 1)$. Definujme nasledujúci pojem.

Definícia 3.2. *Base-m polynóm f_1 je χ-malý, ak* $\max\{|a_i|/m\} \leq \chi$, *kde m je koreň f_1 modulo N.*

Odhadnime veľkosť hodnôt, ktoré nadobúda F_1. Nech U je horná medza $|a|, b$ pre (a, b) z presievacej oblasti. Potom

$$|F_1(a,b)| \leq \sum_{i=0}^{d} \left|a_i a^i b^{d-i}\right| \leq (d+1)\chi m U^d.$$

Je teda jasné, že z hľadiska veľkosti polynomiálnych hodnôt je lepší ten polynóm, ktorý má menšie koeficienty. Pre ilustráciu, pre dostatočne veľké N nie je problém nájsť base-m polynóm s $\chi \approx 1/300$. Hodnoty polynómu F_2 nemusíme uvažovať, pretože $|F_2| \ll |F_1|$ na presievacej oblasti.

Venujme sa teraz voľbe stupňa d. Ako sme spomenuli v závere Oddielu 2.2.3, voľba d sa optimalizuje za účelom minimalizovať zložitosť NFS. Pre N v rozsahu, ktorý nás zaujíma a za predpokladu, že pracujeme s base-m polynómami, sú optimálne hodnoty d uvedené v Tabuľke 3.1 (pre odvodenie pozri [5]). Poznamenajme, že pre čísla dĺžky do 110 cifier je Montgomeryho dva-kvadratická metóda zrejme výhodnejšia. Takisto polynomiálne páry stupňa (4,2) by mohli byť lepšie pre čísla v rozsahu 120–220 cifier. Na vytváranie dobrých polynomiálnych párov takéhoto stupňa ale zatiaľ nie je známy algoritmus.

Počet cifier N	d
80–120	4
120–220	5
220–300	6

Tabuľka 3.1: Optimálna voľba d

3.2.3 Vplyv koreňových vlastností

V tomto oddiele popíšeme model kvantifikácie koreňových vlastností vyvinutý Murphym v [5], tzv. *model typickej F-hodnoty*. Hlavnou myšlienkou modelu je porovnať priemernú p-valuáciu hodnôt nadobúdaných polynómom k priemernej p-valuácii prirodzených čísel. Pripomeňme, že p-valuácia čísla r je exponent najväčšej mocniny prvočísla p, ktorá delí r. Budeme ju značiť $v_p(r)$. Pod pojmom *f-hodnota* pre polynóm f budeme rozumieť hodnotu f v ľubovoľnom celočíselnom bode.

Hodnoty polynómu sa nesprávajú ako náhodné čísla. Ak má polynóm f viac rôznych koreňov modulo p, pravdepodobnosť, že náhodne vybraná f-hodnota je deliteľná p, sa zvýši. Naopak, ak f nemá ani jeden koreň modulo p, potom žiadna f-hodnota nie je deliteľná p.

Aby sme mohli počítať rozdiel medzi priemernou p-valuáciou f-hodnôt a prirodzených čísel, zavedieme nasledujúci pojem:

Definícia 3.3. *Nech $S \subset \mathbb{N}$ a p je prvočíslo. Potom $cont_p(S)$ bude označovať priemernú hodnotu $v_p(r)$ pre r prebiehajúce množinu S. Ak S je množina všetkých g-hodnôt polynómu g, hovoríme o $cont_p(g)$.*

Napríklad pre konečnú množinu S je

$$cont_p(S) = \frac{\sum_{r \in S} v_p(r)}{|S|}.$$

Spočítajme $cont_p(\mathbb{N})$. V množine prirodzených čísel je každé p-té číslo deliteľné číslom p, každé p^2-hé číslo deliteľné p^2, atď. Číslo deliteľné p^2 už raz bolo započítané medzi číslami deliteľnými p, podobne pre vyššie mocniny, preto

$$cont_p(\mathbb{N}) = \sum_{i=1}^{\infty} \frac{1}{p^i} = \frac{1}{p-1}.$$

Zamerajme sa teraz na $cont_p$ pre množinu polynomiálnych hodnôt. Nasledujúci spôsob výpočtu nemôžeme uplatniť pre všetky prvočísla, presnejšie nemôžeme ho použiť pre p, ktoré delí $[\mathbb{Z}_K : \mathbb{Z}[\theta]]$, kde θ je komplexný koreň f.

Spočítajme najskôr $cont_p(f)$. Ak f má jeden koreň modulo p, potom každá p-tá f-hodnota je deliteľná p, keďže

$$f(x) \equiv f(x \bmod p) \pmod{p}.$$

Pre každé $k \geq 2$ odpovedá tento koreň práve jednému koreňu f modulo p^k (Henselova lema). Teda každá p^k-tá f-hodnota je deliteľná p^k. Tým dostávame úvahu totožnú s úvahou vyššie, čiže $cont_p(f) = cont_p(\mathbb{N})$. Ak f má q_p rôznych koreňov modulo p, potom nie je ťažké si rozmyslieť, že

$$cont_p(f) = \frac{q_p}{p-1}.$$

Pre polynóm $F(x,y) = y^{deg\, f} f(x/y)$ je situácia trochu komplikovanejšia. Dvojica (a,b) je koreňom F mod p, ak a/b (uvažované v $\mathbb{Z}_p$) je koreňom $F(x,1) = f(x)$. Ďalšie korene, tzv. *projektívne korene*, vznikajú ak p delí b a vedúci koeficient f. Nech q_p je počet koreňov f mod p, poprípade zvýšené o 1, ak p delí vedúci koeficient f. Potom (viď [5])

$$cont_p(F) = q_p \frac{p}{p^2-1}. \tag{3.2}$$

Pre prvočísla p, pre ktoré nemôžeme využiť vyššie uvedené vzťahy, sa odhad $cont_p$ robí jednoduchým vyčíslením polynómu f na dostatočne veľkej množine $S_1 \subset \mathbb{Z}$. Potom

$$cont_p(f) \approx \frac{\sum_{x \in S_1} v_p(f(x))}{|S_1|}. \tag{3.3}$$

Podobne pre polynóm F

$$cont_p(F) \approx \frac{\sum_{(x,y) \in S_2} v_p(F(x,y))}{|S_2|}, \tag{3.4}$$

pre dostatočne veľkú $S_2 \subset \mathbb{Z} \times \mathbb{Z}$.

Po zavedení pojmu $cont_p$ a jeho spočítaní (resp. odhadnutí) v špeciálnych prípadoch, definujeme pre polynómy f, F cieľ tohto odstavcu, parameter α. Ten bude kvantifikovať rozdiel medzi správaním sa polynomiálnych hodnôt a náhodných čísel rovnakej veľkosti (v závislosti na koreňových vlastnostiach). Parameter α pre f sa

definuje nasledovne:

$$\begin{aligned}\alpha(f) &= \sum_{p\le B} [cont_p(\mathbb{N}) - cont_p(f)] \log p \\ &= \sum_{p\le B} \left[\frac{1}{p-1} - cont_p(f)\right] \log p,\end{aligned}$$

kde B je hranica hladkosti. Pre F je definícia obdobná.

Zamyslime sa nad tým, čo vlastne paramter α znamená. Pre pevne zvolené a dostatočne veľké $r \in \mathbb{N}$ bude pre náhodné hodnoty i_r (Definícia 3.1) ich priemerná p-valuácia približne rovná $cont_p(\mathbb{N})$. Takisto, podľa (3.3) a (3.4), priemerná p-valuácia hodnôt $f(x)$ (resp. $F(x,y)$) na dostatočne veľkej (náhodne zvolenej) množine bude blízko $cont_p(f)$ (resp. $cont_p(F)$). Ak budeme i_r postupne deliť všetkými prvočíslami $p \le B$, v priemere nám na konci ostane v logaritmickom zápise

$$\log i_r - \sum_{p\le B} cont_p(\mathbb{N}) \log p = \log i_r - \sum_{p\le B} \frac{\log p}{p-1}\,. \tag{3.5}$$

Pre polynóm F (môžeme uvážiť aj f) pri rovnakej úvahe dostaneme

$$\log |F(x,y)| - \sum_{p\le B} cont_p(F) \log p\,. \tag{3.6}$$

Uvedomme si, že výraz (3.5) určitým spôsobom charakterizuje pravdepodobnosť, že i_r je B-hladké. V skratke, čím je (3.5) menšie, tým je daná pravdepodobnosť vyššia. Takisto (3.6) charakterizuje pravdepodobnosť $F(x,y)$ byť B-hladkou. Nech teraz $r \approx |F(x,y)|$, potom z definície $\alpha(F)$ a pri použití $\log i_r \approx \log |F(x,y)|$ dostávame

$$\log |F(x,y)| - \sum_{p\le B} cont_p(F) \log p \;\approx\; \log i_r - \sum_{p\le B} \frac{\log p}{p-1} + \alpha(F)\,.$$

Po drobnej úprave máme

$$\log |F(x,y)| - \sum_{p\le B} cont_p(F) \log p \;\approx\; \log \left(i_r e^{\alpha(F)}\right) - \sum_{p\le B} \frac{\log p}{p-1}\,.$$

Tým sme získali vzťah medzi (3.5) a (3.6). Vyplýva z neho, že hodnoty $F(x,y)$ sú B-hladké s približne rovnakou pravdepodobnosťou ako náhodné čísla $e^{\alpha(F)}$-krát väčšie. Môžeme teda povedať, že hodnoty $F(x,y)$ sa správajú ako náhodné čísla veľkosti

$F(x,y)\cdot e^{\alpha(F)}$. Samozrejme, budeme chcieť, aby $\alpha(F)<0$. Ak napríklad $\alpha(F)\approx -7$, potom hodnoty F sú B-hladké s pravdepodobnosťou rovnakou ako pre náhodné čísla, ktoré sú 1000-krát menšie ($e^{-7}\approx 0,001$).

Z definície α je vidieť, že jeho hodnota podstatne závisí od q_p pre malé prvočísla p. Preto ak chceme, aby α bolo záporné, musí mať daný polynóm veľa koreňov modulo malé p.

3.2.4 Odhad kvality polynómu

V predchádzajúcom odstavci sme uvážili, že hodnoty polynómu $f(x)$ sa správajú ako náhodné čísla veľkosti $f(x)e^{\alpha(f)}$. Využitím tohto poznatku a pomocou vlastností Dickmanovej funkcie odhadneme v tomto oddiele počet B-hladkých hodnôt nadobúdaných polynómom f na nejakom intervale.

Označme $P(f(x),B)$ pravdepodobnosť, že náhodne zvolená f-hodnota v absolútnej hodnote menšia ako $|f(x)|$ je B-hladká. Potom z vyššie spomínanej úvahy a s využitím vzťahu (3.1) dostávame

$$P(f(x),B)\approx \rho(u_{f(x)})+\frac{(1-\gamma)\rho(u_{f(x)}-1)}{\log|f(x)|},$$

kde $u_{f(x)}=(\log|f(x)|+\alpha(f))/\log B$.

Nech teraz $I\subset\mathbb{Z}$ je dostatočne veľký interval. Rozdeľme I na disjunktné rovnako veľké podintervaly $I_1,\ldots,I_k$ tak, aby sa hodnoty f na každom I_j extrémne nemenili. Označme ďalej f_j priemernú hodnotu $|f|$ na I_j. Potom počet B-hladkých hodnôt na intervale I môžeme odhadnúť pomocou

$$\frac{|I|}{k}\sum_{j=1}^{k}\rho(u_{f_j})+\frac{(1-\gamma)\rho(u_{f_j}-1)}{\log f_j}. \tag{3.7}$$

Aby sme sa priblížili skutočnému počtu B-hladkých hodnôt na I, musí byť k veľké. V [5] je tento odhad experimentálne overený na siedmych kvadratických polynómoch vytvorených Montgomeryho dva-kvadratickou metódou pre 105, 106 a 107-ciferné čísla. Použitý interval I obsahoval 10^8 bodov a pracovalo sa s $k=10^5$. Dosiahnutá priemerná relatívna chyba bola iba 5,9 %.

Podobným spôsobom, ako sme skonštruovali odhad (3.7) pre f, môžeme skonštruovať aj odhad pre polynóm $F(x,y)$. Rozdiel bude len v tom, že namiesto intervalu I použijeme presievaciu oblasť S, ktorú budeme deliť na podoblasti.

Týmto sme dokončili popis kvantifikácie vlastností polynómu a ich vplyvu na jeho kvalitu. Našou ďalšou úlohou je zistiť, ako konštruovať kvalitné polynómy.

3.3 Generovanie kvalitných polynómov

V nasledujúcej časti sa zameriame na algoritmus, ktorý konštruuje tzv. *neskosené polynómy*[1] (viď [5]). Algoritmus má dve hlavné časti. V prvej je generovaných veľa polynómov a priebežne sa z nich vyberajú tie lepšie. V časti druhej sa z týchto lepších zvolí ten najlepší. Pre jednoduchosť zápisu budeme ďalej používať značenie z Oddielu 3.1.

Polynómy sú v algoritme generované pomocou base-m metódy a to nasledovne. Zvolí sa kladné $a_d \ll N^{1/(d+1)}$ a k nemu vhodné m. Potom sa dopočítajú ostatné koeficienty base-m reprezentácie N:

$$a_i = \left\lfloor \frac{N - \sum_{j=i+1}^{d} a_j m^j}{m^i} \right\rfloor .$$

Následne sa na a_i použije transformácia spomenutá v Oddiele 3.1.

Samozrejme, nechceme takto konštruovať ľubovoľné base-m polynómy. Potrebujeme, aby mali dobré vlastnosti z hľadiska ich kvality. Prvou požiadavkou je, aby boli χ-malé pre nami zvolené χ. Od a_d teda požadujeme

$$a_d < \chi m,$$

$$-m^d < N - a_d m^d < m^d.$$

Kombináciou týchto vzťahov a použitím $a_d - 1 \approx a_d$ dostávame hornú hranicu pre a_d

$$a_d < \left(\chi^d N\right)^{\frac{1}{d+1}} \tag{3.8}$$

Ďalšou nutnou podmienkou na konštruovaný polynóm, aby bol χ-malý, je platnosť vzťahu $|a_{d-1}|/m \leq \chi$, t.j.

$$-\chi m \leq \frac{N - a_d m^d}{m^{d-1}} \leq \chi m.$$

Potom pre m musí platiť

$$\left(\frac{N}{a_d + \chi}\right)^{\frac{1}{d}} \leq m \leq \left(\frac{N}{a_d - \chi}\right)^{\frac{1}{d}}. \tag{3.9}$$

Pre N veľké to ani zďaleka nie je obmedzujúca podmienka. Už pre 70-ciferné N, $\chi = 1/2500$ a vhodné a_d (v tomto prípade je $d = 3$) má interval vymedzený nerovnosťami (3.9) približne 10^8 bodov.

[1] z angl. *non-skewed polynomials*

Zamerajme sa teraz na koreňové vlastnosti polynómu F_1. Algoritmus využíva jednoduchú myšlienku ako ich zlepšiť vhodnou voľbou a_d. Ak vedúci koeficient a_d je deliteľný malými prvočíslami, zvýši sa tým počet projektívnych koreňov F_1 a teda hodnota $\alpha(F_1)$ bude menšia.

Poznámka 3.1. Na nasledujúcom experimente urobenom v [5] ukážeme, aký efekt môže mať vhodná voľba a_d. Experiment bol vykonaný na 140-cifernom čísle. Pre dané N bol vedúci koeficient a_d vyberaný z intervalu $[10^{20,3}, 10^{21,3}]$ nasledovne:

1. a_d je prvočíslo,
2. a_d je volené náhodne,
3. $c|a_d$ pre $c = 2 \cdot 3 \cdot 5 \cdot 7$,
4. $c|a_d$ pre $c = 2^5 \cdot 3^4 \cdot 5^3 \cdot 7^3 \cdot 11^2 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \approx 10^{16,6}$.

V každom z prípadov uvedených vyššie bolo náhodne vygenerovaných sto polynómov a následne sa spočítali minimum, maximum a priemerné hodnoty $\alpha(F_1)$. Výsledky sú v Tabuľke 3.2.

Prípad č.	Min $\alpha(F_1)$	Priemer $\alpha(F_1)$	Max $\alpha(F_1)$
1	-0,67	1,16	2,63
2	-1,62	0,44	2,33
3	-3,02	-0,77	0,77
4	-3,54	-1,88	-0,47

Tabuľka 3.2: Hodnoty $\alpha(F_1)$ pre rôzne a_d

Nasleduje popis algoritmu na konštrukciu neskosených polynómov. V popise spomenuté χ, α_{max} a K sú pevne dané konštanty.

Algoritmus 3.2 (Neskosené polynómy)**.** 1. (a) Zvoľ a_d, ktoré spĺňa (3.8) a zároveň je deliteľné veľkým počtom malých prvočísel. Pokračuj na (1b).

(b) Vyber m z intervalu určeného pomocou (3.9). Spočítaj base-m reprezentáciu. Ak je daný polynóm χ-malý, prejdi na krok (1c), inak zopakuj (1b).

(c) Spočítaj odhad $\alpha(F_1)$. Ak $\alpha(F_1) \leq \alpha_{max}$, polynóm postupuje do druhej fázy.

Krok (1b) a (1c) opakujeme, kým pre aktuálne a_d nenájdeme dostatočný počet polynómov postupujúcich do druhej fázy. Potom sa môžeme vrátiť späť na krok (1a). Keď celkový počet nájdených vhodných polynómov je rovný K, ukončíme hľadanie a prejdeme na krok (2).

2. Ohodnoť polynómy, ktoré postúpili do druhej fázy a vyber polynóm s najlepším ohodnotením.

Spôsobu, akým sa polynómy v druhej časti algoritmu ohodnocujú, sa budeme venovať nižšie v Odstavci *Ohodnotenie polynómov*. V Odstavci *Metódy výpočtu* α popíšeme, ako sa odhaduje hodnota $\alpha(F_1)$.

Rozhodujúci vplyv na dobu behu algoritmu a kvalitu výsledného polynómu majú parametre, ktoré rozhodujú o postupe konštruovaných polynómov do druhej časti algoritmu. Konkrétne sú to χ a α_{max}. Ak ich nastavenie bude príliš „tolerantné“, algoritmus síce dobehne rýchlo, ale kvalita výsledného polynómu bude nízka. Naopak, ak ich nastavíme prísne, môže sa stať, že žiaden polynóm nepostúpi do druhej fázy. V Kapitole 4 sa budeme venovať optimálnemu nastaveniu týchto konštánt pre daný čas t (doba behu algoritmu). Pomocou experimentov odvodíme optimálne hodnoty χ a α_{max}, pri ktorých za čas t nájdeme v porovnaní s inými nastaveniami χ' a α'_{max} najlepšie neskosené polynómy.

Ďalším faktorom majúcim vplyv na výpočtovú dobu Algoritmu 3.2 a kvalitu výsledku je K, t.j. počet polynómov, ktoré chceme nazbierať do druhej časti. Je jasné, že ak je K väčšie, budeme vyberať výsledný polynóm z väčšej množiny a jeho kvalita nemôže byť horšia. Na druhej strane, výpočtová doba sa zväčšením K zvýši. Vplyvom tohto parametru sa taktiež budeme zaoberať.

Poznamenajme, že v [5] je okrem tohto algoritmu vyvinutý ďalší, ktorý konštruuje ešte kvalitnejšie polynómy, tzv. *skosené polynómy*[2].

Ohodnotenie polynómov

Medzi polynómami, ktoré postúpili do druhej fázy Algoritmu 3.2, stále môže byť významný rozdiel v ich kvalite. Jednou z možností, ako vybrať z nich ten najlepší, je vykonať s každou dvojicou (F_1, F_2) (príslušnou k polynomiálnemu páru (f_1, f_2)) presievací experiment a na základe toho sa rozhodnúť. Keďže je ale polynómov v druhej fáze stále veľa, bola by to časovo náročná operácia. Preto potrebujeme rýchlu metódu na ohodnotenie kvality každého polynómu.

[2] z angl. *skewed polynomials*

Všimnime si, že pre polynómy F_1, F_2, s ktorými sa pracuje v NFS, platí

$$F_i(cx, cy) = c^{\deg f_i} F_i(x, y).$$

Potom pre (x, y) v polárnych súradniciach dostávame

$$F_i(x, y) = r^{\deg f_i} F(\cos\theta, \sin\theta).$$

Vieme, že počet B-hladkých hodnôt nadobúdaných polynómom F na nejakej množine výrazne závisí od veľkosti nadobúdaných hodnôt a parametru $\alpha(F)$. Navyše, ak chceme skúmať rozdiel vo veľkosti nadobúdaných hodnôt dvoch polynómov F, G ($\deg f = \deg g$) na nejakej množine, vďaka vyššie uvedenému vzťahu stačí, keď sa obmedzíme na ich správanie sa na jednotkovej kružnici. To využijeme v nasledujúcej metóde. Poznamenajme ešte, že presievacia oblasť S je vždy podmnožinou $\mathbb{Z} \times \mathbb{N}$. Preto sa môžeme obmedziť na polkružnicu nachádzajúcu sa nad osou x.

Pre $K \in \mathbb{N}$ (v [5] sa volí $K = 1000$) položme

$$\theta_i = \frac{\pi}{K}\left(i - \frac{1}{2}\right),\ i = 1, \ldots, K.$$

Definujme *ohodnotenie polynómu* F

$$\mathbb{E}(F) = \sum_{i=1}^{K} \rho\left(\frac{\log|F(\cos\theta_i, \sin\theta_i)| + \alpha(F)}{\log B}\right).$$

V praktickom použití sa ohodnotenie F môže počítať na polkružnici s polomerom $R > 1$, aby sa zabránilo prípadom, keď výraz vo vnútri Dickmanovej funkcie je menší ako 1.

Z definície Dickmanovej funkcie a úvah v Odstavci 3.2.3 môžeme ohodnotenie F interpretovať nasledovne. Hodnota $\mathbb{E}(F)/K$ je odhadom asymptotickej pravdepodobnosti, že $F(x, y)$ je B-hladké pre (x, y) blízko polkružnice s polomerom 1 (alebo R).

Podobne môžeme definovať aj ohodnotenie dvojice (F_1, F_2) príslušnej k polynomiálnemu páru (f_1, f_2). Hranica hladkosti môže byť pre obidva polynómy rôzna, preto označíme B_{F_j} hranicu hladkosti príslušnú k F_j, $j = 1, 2$. Ďalej označme

$$u_{F_j}(\theta) = \frac{\log|F_j(\cos\theta, \sin\theta)| + \alpha(F_j)}{\log B_{F_j}}, \quad j = 1, 2.$$

Potom sa *ohodnotenie* (F_1, F_2) definuje nasledovne

$$\mathbb{E}(F_1, F_2) = \sum_{i=1}^{K} \rho\left(u_{F_1}(\theta_i)\right) \rho\left(u_{F_2}(\theta_i)\right).$$

Hodnotu $\mathbb{E}(F_1, F_2)/K$ môžeme opäť interpretovať ako odhad asymptotickej pravdepodobnosti, že $F_1(x, y)$ je B_{F_1}-hladké a zároveň $F_2(x, y)$ je B_{F_2}-hladké pre (x, y) blízko polkružnice s polomerom 1 (alebo R).

Tým sme dokončili popis metódy na ohodnotenie polynómov. Je ale dôležité uvedomiť si, že porovnávať $\mathbb{E}(F)$ a $\mathbb{E}(G)$, resp. $\mathbb{E}(F_1, F_2)$ a $\mathbb{E}(G_1, G_2)$, má zmysel len pre $\deg f = \deg g$, resp. $(\deg f_1, \deg f_2) = (\deg g_1, \deg g_2)$.

V Algoritme 3.2 na konštrukciu neskosených polynómov sa pomocou tejto metódy ohodnotia všetky polynómy, ktoré postúpili do druhej fázy. Potom z nich vyberieme najlepšie ohodnotený. Druhou možnosťou je, že sa vyberie niekoľko polynómov s najlepšími ohodnoteniami a na základe presievacích experimentov z nich zvolíme ten najlepší.

Metódy výpočtu α

Parameter $\alpha(F)$ sa v praxi nepočíta presne. Suma, ktorou je definovaný, je príliš veľká a navyše nevieme, ktoré p delí $[\mathbb{Z}_K : \mathbb{Z}[\theta]]$ a ktoré nie. To nám bráni v použití vzťahu (3.2).

Rýchly ale hrubý odhad $\alpha(F)$ môžeme počítať pomocou

$$\alpha(F) \approx \sum_{p<P} \left(1 - q_p \frac{p}{p+1}\right) \frac{\log p}{p-1},$$

kde P je nami zvolená hodnota[3]. Tento spôsob odhadu sa používa v kroku (1c) Algoritmu 3.2, kde sa počíta $\alpha(F)$ pre veľa polynómov.

Pri počítaní $\mathbb{E}(F)$ v Algoritme 3.2 potrebujeme presnejší odhad $\alpha(F)$, aby sme vybrali skutočne najkvalitnejšie polynómy. Naviac môžeme venovať odhadu $\alpha(F)$ viac času. Preto na výpočet $cont_p(F)$ použijeme pre $p < P_1$ vzťah (3.4), ktorý je presnejší. Ďalej pre p spĺňajúce $P_1 \leq p < P_2$ položíme $cont_p(F) = q_p \frac{p}{p^2-1}$. To všetko pre nami zvolené P_1, P_2.[4] Výsledný odhad potom bude

$$\alpha(F) = \sum_{p<P_2} \left(\frac{1}{p-1} - cont_p(F)\right) \log p\,.$$

[3] V implementácii algoritmu, s ktorou budeme pracovať v Kapitole 4 je $P = 100$.

[4] V implementácii algoritmu, s ktorou budeme pracovať v Kapitole 4 je $P_1 = 100$ a $P_2 = 2000$.

Kapitola 4

Hľadanie optimálnych konštant pre generovanie polynómov

Ako sme už spomenuli v Odstavci 3.3, výsledok a doba behu Algoritmu 3.2 závisia na nastavení parametrov α_{max}, χ a K. Našou úlohou v tejto kapitole bude pre pevné K na základe experimentov nájsť optimálne konštanty α_{opt} a χ_{opt}. Presnejšie, pre čas t chceme nájsť $\alpha_{opt}(t)$, $\chi_{opt}(t)$ spĺňajúce:

1. pre dané parametre (K je pevne dané) je v priemernom prípade doba výpočtu (behu algoritmu) približne t,

2. ohodnotenie výsledného polynómu bude v priemernom prípade lepšie ako ohodnotenie výsledného polynómu získaného pri inom nastavení α', χ' spĺňajúcom bod č. 1.

Množinu bodov (α', χ') spĺňajúcich bod č. 1 nazveme *izochronou pre čas t*.

Optimálne nastavenia α_{max} a χ sme hľadali pre polynomiálne páry stupňa (3,1), (4,1) a (5,1). Na záver kapitoly budeme skúmať vplyv parametru K na kvalitu výsledku.

Pri experimentoch sme pracovali s implementáciou Algoritmu 3.2, ktorá bola naprogramovaná na Katedre algebry Matematicko-fyzikálnej fakulty Karlovej univerzity.

4.1 Popis experimentu

V tomto oddiele podrobnejšie popíšeme, ako prebiehali experimenty zamerané na hľadanie optimálneho nastavenia konštánt α_{max}, χ pri pevne zvolenom K.

Pre polynomiálne páry stupňa (3,1) sme robili výpočty so 60 a 70-cifernými číslami, ďalej pre polynomiálne páry stupňa (4,1) s 80 a 90-cifernými číslami a pre páry stupňa (5,1) so 130-cifernými číslami. Vo všetkých prípadoch sme sa rozhodli pre nastavenie parametru $K = 100$.

Pre každý stupeň a číselný rád je postup experimentu totožný. Môžeme ho rozdeliť na dve fázy: *inicializačnú* a *hlavnú*. V inicializačnej fáze sa vygenerujú čísla $N_1, N_2, \ldots, N_{10}$ príslušného rádu, pričom každé N_i je súčinom dvoch rôznych náhodne vygenerovaných prvočísel rovnakej dĺžky. Potom sa hľadajú nastavenia parametrov α_{max} a χ, s ktorými budeme spúšťať výpočty, nasledovne:

- pre zvolený stupeň určíme vhodné nastavenia α_{max}, označme ich $\alpha_1, \ldots, \alpha_n$; voľbu $\alpha_1, \ldots, \alpha_n$ špecifikujeme neskôr v Oddiele 4.2
- pre každé α_i položíme $\chi_{i,1} = 1/2$ a experimentálne odvodíme $\chi_{i,15}$ tak, aby výpočet s parametrami α_i, $\chi_{i,15}$, $K = 100$ a náhodne vybranými číslami N_j trval v priemere približne 40 minút[1]; $\chi_{i,2}, \ldots, \chi_{i,14}$ sa vyberú približne rovnomerne z intervalu $(\chi_{i,15}, \chi_{i,1})$.

V hlavnej fáze pre každé N_i spustíme Algoritmus 3.2 s nastavenými parametrami

$$\alpha_j,\ \chi_{j,l},\ K = 100, \quad l = 1, \ldots, 15,\ j = 1, \ldots, n.$$

Zaujíma nás doba výpočtu a výsledné ohodnotenie $\mathbb{E}(F_1)$.

Spracovanie výsledkov

Zo zozbieraných údajov urobíme aritmetický priemer. Pomocou týchto dát skonštruujeme pre vybrané časy $\tau_1, \ldots, \tau_k$ izochrony $I_1, \ldots, I_k$. Na každej izochrone nájdeme optimálne nastavenie $\alpha_{opt}(\tau_i)$, $\chi_{opt}(\tau_i)$. Získané optimálne nastavenia budeme diskutovať v Oddiele *Výsledky experimentov*.

Konštrukcia izochrón:

1. Pre každé α_i aproximujeme funkciou $T_{\alpha_i}(\chi)$ výpočtové časy algoritmu s parametrami $(\alpha_i, \chi_{i,l})$, $l = 1, \ldots, 15$. Podobne aproximujeme funkciou $R_{\alpha_i}(\chi)$ hodnoty $\mathbb{E}(F_1)$.

   Hodnota $T_{\alpha_i}(\chi)$ teda predstavuje očakávanú dĺžku výpočtu pre parametre α_i, χ $(K = 100)$. $R_{\alpha_i}(\chi)$ je očakávané ohodnotenie polynómu získaného pre dané parametre.

[1] Výnimkou je polynomiálny pár stupňa (5,1), kde je horná časová hranica päť hodín.

2. Pre každé α_i a každé τ_j nájdeme $\psi_{i,j}$ (nastavenie parametru χ) tak, aby platilo $T_{\alpha_i}(\psi_{i,j}) = \tau_j$. Bod $(\alpha_i, \psi_{i,j})$ pridáme do izochrony I_j. Zároveň spočítame $R_{\alpha_i}(\psi_{i,j})$, t.j. ohodnotenie polynómu, ktorý získame výpočtom s parametrami $\alpha_i, \psi_{i,j}, K = 100$.

   Nie vždy musí $\psi_{i,j}$ existovať. Ak neexistuje alebo nepatrí do intervalu $(0, 1/2)$ (iné hodnoty nedávajú rozumný zmysel), znamená to, že pre dané α_i je výpočetný čas pri ľubovoľnej voľbe parametru χ väčší ako τ_j (pre $K = 100$). Na druhej strane, ak je takýchto vzorov viac, vyberieme vždy najmenší z intervalu $(0, 1/2)$.

Na každej izochrone aproximujeme správanie sa $\mathbb{E}(F_1)$ pomocou hodnôt $R_{\alpha_i}(\psi_{i,j})$, $i = 1, \ldots, n$. Potom nájdeme na každej I_j bod, kde sa nadobúda maximum $\mathbb{E}(F_1)$, t.j. optimálne nastavenie parametrov (α_{max}, χ) pre čas τ_j a $K = 100$.

4.2 Výsledky experimentov

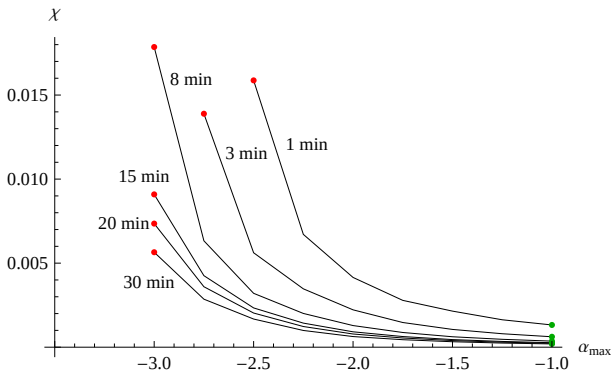
V oddieloch 4.2.1 až 4.2.3 budeme prezentovať výsledky experimentov zameraných na hľadanie optimálnych nastavení α_{max} a χ. Keďže pri daných výpočtoch bolo $K = 100$, pri diskusii o voľbe parametrov v týchto oddieloch budeme vždy predpokladať toto nastavenie. Nakoniec sa v Oddiele 4.2.4 zameriame na vplyv parametru K na kvalitu výsledku.

Poznámka 4.1. Venujme sa chvíľu úvahe, ako sa zmení optimálne nastavenie α_{max} a χ, ak zvýšime K z hodnoty 100 na nejaké K_1. Pri takejto voľbe môžeme celkom prirodzene očakávať, že výpočtový čas bude $K_1/100$-krát väčší v porovnaní s nastavením $K = 100$ a rovnakými α_{max}, χ (vplyvu K na výpočtový čas sa budeme venovať v Oddiele 4.2.4). Teda nastavenia α_{max}, χ, ktoré ležali na rovnakej izochrone pri nastavení $K = 100$, by mali pri zvýšení K opäť ležať na jednej izochrone a zároveň na tejto izochrone k nim nepribudne žiadna nová kombinácia nastavenia α_{max}, χ. Preto sa optimálne nastavenia pri zvýšení K výrazne nezmenia, zmení sa len výpočtový čas, pre ktorý sú tieto nastavenia optimálne.

4.2.1 Polynomiálne páry stupňa (3,1)

Pre polynomiálne páry stupňa $(3, 1)$ sme nastavenia parametru α_{max} vyberali z množiny $\{-1; -1,25; -1,5; \ldots; -3\}$. Pre nižšie voľby α_{max} bežal výpočet pri ľubovoľných nastaveniach χ viac ako 30 minút, preto sme sa o nižšie hodnoty α_{max} nezaujímali.

Polynómy s najvyšším ohodnotením boli na skonštruovaných izochronách nájdené vždy pri nastavení parametru $\alpha_{max} = -1$ (a príslušnom χ). Naopak, najhoršie polynómy boli skonštruované pri $\alpha_{max} = -3$. Na Obrázku 4.1 sú znázornené izochrony pre 70-ciferné čísla pre časy 1, 3, 8, 15, 20 a 30 minút. Zelený bod na izochrone odpovedá nastaveniu, pri ktorom je nájdený polynóm s najlepším ohodnotením v porovnaní s ostatnými bodmi izochrony. Naopak, červený bod odpovedá nastaveniu s najhorším výsledkom spomedzi bodov izochrony. Poznamenajme, že izochrony pre časy 1 a 3 minúty neobsahujú body s príliš nízkym α_{max}. To odpovedá faktu, že pre dané α_{max} nie je možné dosiahnuť (pri ľubovolnej voľbe χ) tak nízky výpočtový čas. (Pri týchto izochronách sa polynómy s najnižším $\mathbb{E}(F_1)$ vyskytli pri $\alpha_{max} = -2.5$ pre čas 1 minúta, resp. pri voľbe $\alpha_{max} = -2,75$ pre čas 3 minúty.)

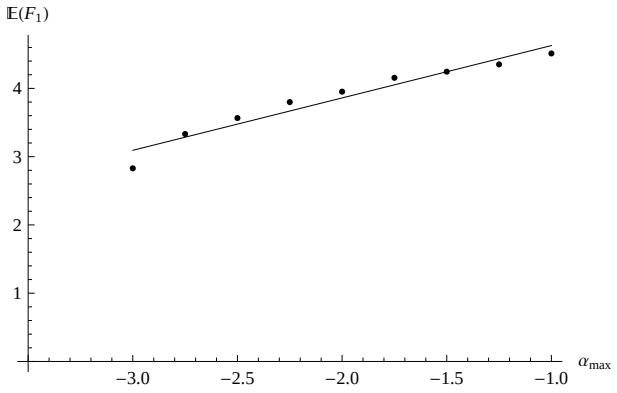


Obr. 4.1: Izochrony, 70-ciferné čísla

Je dôležité zvýrazniť, že na každej izochrone sa $\mathbb{E}(F_1)$ výrazne mení, t.j. za rovnaký čas môžeme pri vhodnej voľbe (α_{max}, χ) nájsť významne lepší polynóm. Presnejšie, priemerná hodnota $\mathbb{E}(F_1)$ v „najlepšom“ bode izochrony je 1,4 až 1,6-krát väčšia ako priemerná hodnota v jej „najhoršom“ bode. Obrázok 4.2 znázorňuje „správanie sa“ $\mathbb{E}(F_1)$ na izochrone pre čas 8 minút (skonštruovanej z výsledkov experimentov pre 60-ciferné čísla). Ako môžeme vidieť, závislosť $\mathbb{E}(F_1)$ na α_{max} je takmer lineárna.[2] Podobne sa $\mathbb{E}(F_1)$ správa aj na ostatných izochronách pre 60 a 70-ciferné čísla.

Z experimentov vyplýva, že za rovnaký čas sme našli najkvalitnejšie polynómy pri najväčšom zvolenom $\alpha_{max}(= -1)$ a najhoršie polynómy pri najmenšom α_{max}. Na

[2]V skutočnosti $\mathbb{E}(F_1)$ závisí na α_{max} a χ ($K = 100$). V tomto prípade ale skúmame správanie sa $\mathbb{E}(F_1)$ na izochrone, teda ku každému α_{max} je jednoznačne priradené χ (viď odstavec *Spracovanie výsledkov* v Oddiele 4.1). Preto môžeme hovoriť o závislosti len na α_{max}.



Obr. 4.2: $\mathbb{E}(F_1)$ na izochrone pre $t = 8$ minút, 60-ciferné čísla

vysvetlenie výsledku budeme potrebovať nasledujúcu krátku úvahu.

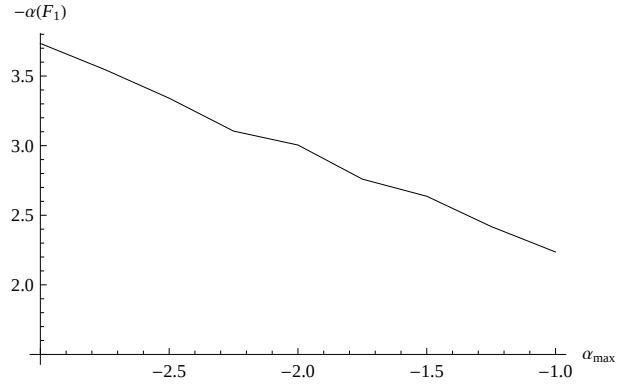
Nech $(f_1, f_2), (g_1, g_2)$ a k ním príslušné $(F_1, F_2), (G_1, G_2)$ sú polynomiálne páry vytvorené base-m metódou k faktorizovanému číslu N. Nech ďalej f_1 je χ_1-malý, g_1 je χ_2-malý. Vieme, že hodnoty $F_1(x, y)$, resp. $G_1(x, y)$, sa správajú ako náhodné čísla veľkosti $|F_1(x, y)|\, e^{\alpha(F_1)}$, resp. $|G_1(x, y)|\, e^{\alpha(G_1)}$ (z hľadiska pravdepodobnosti byť B-hladkými, viď Oddiel 3.2.3). Ich kvalitu môžeme porovnať podielom

$$\frac{|F_1(x, y)|\, e^{\alpha(F_1)}}{|G_1(x, y)|\, e^{\alpha(G_1)}} \approx \frac{\chi_1}{\chi_2} \cdot e^{\alpha(F_1)-\alpha(G_1)}. \tag{4.1}$$

Pre pevnú dobu výpočtu môžeme pri voľbe $\alpha_{max} = -1$ v Algoritme 3.2 voliť χ oveľa menšie ako pri voľbe $\alpha_{max} = -3$. Na skonštruovaných izochronách pre 60 a 70-ciferné čísla je χ 12 až 50-krát menšie pri $\alpha_{max} = -1$. Otázkou ostáva, ako ovplyvní prísnejšia voľba α_{max} koreňové vlastnosti výsledného polynómu.

Na Obrázku 4.3 je znázornený priemerný zisk v koreňových vlastnostiach polynómov získaných pre rôzne α_{max} pri experimentoch na 60-ciferných číslach. Takéto správanie α môžeme očakávať na každej izochrone, hodnoty α totiž nezávisia na voľbe χ. Takmer totožné hodnoty nadobúda α aj pri 70-ciferných číslach, keďže sa pri nich konštruovali polynómy rovnakého stupňa ako pri 60-ciferných číslach.

Rozdiel v koreňových vlastnostiach polynómov nájdených pre nastavenia $\alpha_{max} = -1$ a $\alpha_{max} = -3$ je pomerne významný (rozdiel hodnôt α je približne 1,5), ale použitím vzťahu (4.1) dostávame, že výnos získaný lepšími koreňovými vlastnosťami neprekryje výnos získaný voľbou extrémne malého χ. S α_{max} rastúcim k -1 a zmenšujúcim sa χ (pri zachovaní doby výpočtu) sa kvalita získaných polynómov blíži



Obr. 4.3: α_{max} versus α, 60-ciferné čísla

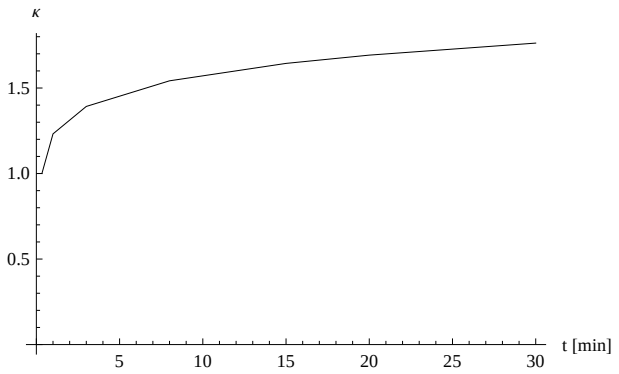
ku kvalite polynómov získaných pre $\alpha_{max} = -1$ (a príslušné χ), viď Obrázok 4.2. Naskytuje sa otázka, či pre $\alpha_{max} > -1$ nedosiahneme ešte lepšie výsledky.

Pri experimentoch vykonaných s nastavením $\alpha_{max} = 0$ je možné v porovnaní s $\alpha_{max} = -1$ voliť χ 1,4–2,6-krát menšie (pre časy 1, 3, 8, 15, 20 a 30 minút). Rozdiel v koreňových vlastnostiach nie je veľký, hodnota α stúpla v priemere iba o dve desatiny. Kvalita výsledných polynómov stúpla už len mierne, hodnota $\mathbb{E}(F_1)$ sa zvýšila na skonštruovaných izochronách v porovnaní s $\alpha_{max} = -1$ o menej ako 7 %.

Vďaka vhodnej voľbe vedúcich koeficientov v Algoritme 3.2 môžeme predpokladať, že pre všetky generované polynómy je $\alpha < 0$ (Poznámka 3.1). Malý rozdiel v hodnotách α pri $\alpha_{max} = 0$ a $\alpha_{max} = -1$ svedčí o tom, že väčšina konštruovaných polynómov má pomerne dobré koreňové vlastnosti. Navyše pri voľbe $\alpha_{max} = 0$ môžeme nastaviť χ výrazne menšie v porovnaní s $\alpha_{max} = -3$, čo prevýši výnos získaný lepšími koreňovými vlastnosťami. Preto je optimálnou voľbou pre konštrukciu polynomiálnych párov stupňa $(3, 1)$ nepočítať odhad $\alpha(F_1)$ a zamerať sa iba na hľadanie χ-malých polynómov s najmenším možným χ pre daný čas.

Na Obrázku 4.4 je znázornená závislosť ohodnotenia výsledného polynómu na dobe výpočtu pri nastavení $\alpha_{max} = 0$ (s meniacou sa dobou výpočtu sa mení χ tak, aby výpočet trval požadovaný čas; ak je takých χ viac, vyberie sa najmenšie). Hodnota κ (zvislá os) je podielom ohodnotenia výsledného polynómu pre daný čas a ohodnotenia polynómu získaného pri $\alpha_{max} = 0$ a dĺžke výpočtu 20 sekúnd. Obrázok 4.4 teda znázorňuje, aké zlepšenie v kvalite polynómov môžeme očakávať pre daný výpočtový čas v porovnaní s dobou výpočtu 20 sekúnd.

V Tabuľke 4.1 sú uvedené nastavenia χ a relatívny nárast $\mathbb{E}(F_1)$ pre vybrané časy, parameter $\alpha_{max} = 0$. Zaujímavosťou je, že pre $\chi = 1/8700$ sa pre 60-ciferné čísla výpočtový čas pohyboval nad hranicou jednej hodiny, čo je dvojnásobok v po-



Obr. 4.4: Relatívny nárast $\mathbb{E}(F_1)$ pre $\alpha_{max} = 0$, 70-ciferné čísla

rovnaní so 70-cifernými číslami (viď Tabuľka 4.1). Pre 70-ciferné čísla bol dokonca pri voľbe $\chi = 1/11000$ priemerný výpočtový čas iba 45 minút. Pri 60-ciferných číslach pre takúto voľbu parametrov musel byť výpočet po niekoľkých hodinách prerušený.

čas	60 cif.		70 cif.	
(min)	$1/\chi$	κ	$1/\chi$	κ
1/3	500	1,00	500	1,00
3	2900	1,44	2600	1,39
8	4800	1,60	4400	1,54
30	6500	1,70	8700	1,76

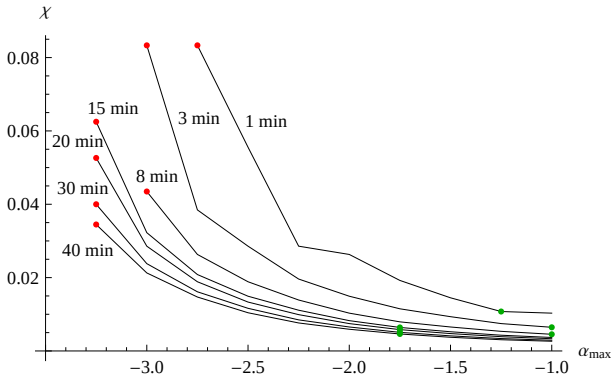
Tabuľka 4.1: Nastavenie χ a relatívny nárast $\mathbb{E}(F_1)$, $\alpha_{max} = 0$

4.2.2 Polynomiálne páry stupňa (4,1)

Pri experimentoch s polynomiálnymi pármi stupňa (4,1) došlo k miernemu poklesu výpočtového času pri nastavení $\alpha_{max} = -3,25$. Preto sme spúšťali výpočty aj s takýmto nastavením α_{max}. Dôvodom spomínaného poklesu je zlepšenie koreňových vlastností generovaných polynómov, ktorému sa podrobnejšie budeme venovať nižšie.

Na izochronách boli polynómy s najnižším ohodnotením opäť nájdené pre najmenšie α_{max}. Situácia sa mierne zmenila pre „najlepšie“ body izochrony. Pre 80-ciferné čísla boli polynómy s najvyšším ohodnotením nájdené pri voľbe $\alpha_{max} = -1,5$,

pre 90-ciferné čísla je situácia znázornená na Obrázku 4.5 (červené a zelené body majú ten istý význam ako v Obrázku 4.1).



Obr. 4.5: Izochrony, 90-ciferné čísla

Pre rôzne nastavenia (α_{max}, χ) ležiace na jednej izochrone nie je rozdiel v kvalite nájdených polynómov taký významný ako v prípade polynomiálnych párov stupňa (3,1) (Oddiel 4.2.1). Na skonštruovaných izochronách stúpla priemerná hodnota $\mathbb{E}(F_1)$ v „najlepšom“ bode izochrony v porovnaní s priemernou hodnotou $\mathbb{E}(F_1)$ v „najhoršom“ bode izochrony o 8 až 17 %. Menší rozdiel je zapríčinený zmenou voľby χ potrebnou k dosiahnutiu požadovaného výpočtového času. Tej sa budeme venovať nižšie, najskôr sa pozrieme na zmeny hodnoty α.

Koreňové vlastnosti výsledných polynómov sa zlepšili oproti polynómom stupňa 3. Keďže pracujeme s polynómami vyššieho stupňa, tie môžu mať viac koreňov modulo prvočíslo p. Tým pádom aj hodnota $\alpha(F_1)$ môže byť nižšia. Tá klesla rovnomerne, pre každé α_{max} v priemere o hodnotu 0,2 v porovnaní s polynómami stupňa 3. Teda rozdiel v koreňových vlastnostiach polynómov získaných pre rôzne nastavenia α_{max} sa zachoval.

Na druhej strane, parameter χ sme museli voliť výrazne vyšší ako pri polynomiálnych pároch stupňa (3,1). To je očakávaný jav. Totiž, pri konštruovaní χ-malého polynómu vhodnou voľbou vedúceho koeficientu a_d a koreňa m v Algoritme 3.2 je zaručené

$$|a_d| \leq \chi m, \quad |a_{d-1}| \leq \chi m.$$

Pre polynóm tretieho stupňa nám ostávajú potom už iba 2 koeficienty, ktoré môžu χ-malosť „pokaziť“. Pri polynómoch štvrtého stupňa sú to až 3 koeficienty. Intuitívne je teda jasné, že nájsť χ-malý polynóm je pre polynómy 4. stupňa ťažšia úloha.

Zmena χ nebola rovnaká pre každé α_{max}. Zatiaľčo pri nastavení $\alpha_{max} = -3$ sme museli voliť χ dva až päťkrát väčšie (pre dané časy) v porovnaní s voľbou χ pre po-

čas	80 cif.			90 cif.		
(min)	α_{max}	$1/\chi$	κ	α_{max}	$1/\chi$	κ
1/3	-1,5	30	1,00	-1,25	30	1,00
3	-1,5	110	1,22	-1,00	160	1,25
8	-1,5	160	1,29	-1,00	220	1,32
20	-1,5	230	1,36	-1,75	170	1,39
40	-1,5	280	1,40	-1,75	220	1,44

Tabuľka 4.2: Optimálne nastavenie parametrov α_{max}, χ a relatívny nárast $\mathbb{E}(F_1)$ pre vybrané časy

lynómy 3. stupňa a rovnakom nastavení α_{max}, pre $\alpha_{max} = -1$ to bol 6 až 15-násobok. V tomto zmysle bola zmena χ výraznejšia pre väčšie α_{max}. Dôsledkom toho je, že pri $\alpha_{max} = -1$ je parameter χ len 7 až 16-krát menší ako pri najmenšom α_{max} (a príslušnom χ) ležiacom na tej istej izochrone. Menší rozdiel vo voľbe χ a pritom zachovanie rozdielu v koreňových vlastnostiach polynómov získaných pre rôzne nastavenia ležiace na tej iste izochrone vysvetľuje menší rozdiel v kvalite získaných polynómov za daný čas.

Voľba optimálnych parametrov už nie je taká jednoznačná, ako v prípade polynomiálnych párov stupňa (3,1). Už nemôžeme konštruovať χ-malé polynómy s extrémne malým χ. Nieže by neexistovali, ale na ich nájdenie by sme potrebovali oveľa viac času. Aj keď výhoda voľby menšieho χ pri väčšom α_{max} už nie je taká výrazná, stále sa neoplatí voliť nízke α_{max} ($\alpha_{max} < -2$). Optimálnou voľbou je zvoliť α_{max} z intervalu $[-2, -1]$ a k nemu určiť χ tak, aby výpočet trval nami požadovanú dobu.

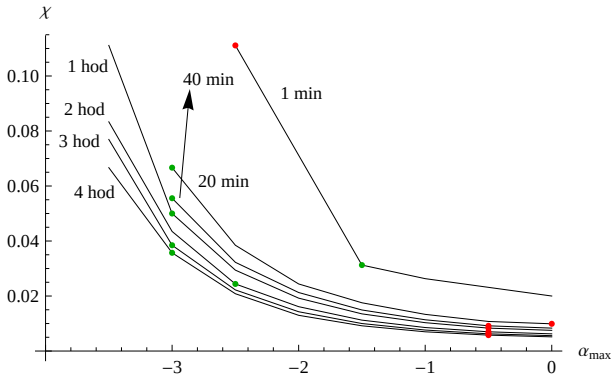
V Tabuľke 4.2 sú uvedené optimálne body skonštruovaných izochrón pre vybrané časy. Hodnota κ vyjadruje relatívny nárast $\mathbb{E}(F_1)$, t.j. podiel ohodnotenia polynómu získaného pre daný čas (a parametre) a najlepšieho ohodnotenia získaného na izochrone pre dobu výpočtu 20 sekúnd. Pri porovnaní s Tabuľkou 4.1 môžeme vidieť, že hodnota $\mathbb{E}(F_1)$ s rastúcim časom stúpa pomalšie ako pri polynomiálnych párach stupňa (3,1).

4.2.3 Polynomiálne páry stupňa (5,1)

Pri experimentoch s polynomiálnymi pármi stupňa (5,1) sme volili parameter α_{max} z množiny $\{0; -0,5; -1; \ldots; -3,5\}$. Pri voľbe $\alpha_{max} = -4$ a ľubovoľnom χ presaho-

vala doba výpočtu hranicu desať hodín, preto sme takéto a nižšie nastavenia α_{max} nebrali do úvahy.

Výpočty prebiehali na 130-ciferných číslach. Faktorizovať čísla takejto veľkosti je náročná úloha. Fáze Hľadanie polynómov sa venuje oveľa viac výpočtového času ako v predchádzajúcich prípadoch. Preto sme konštruovali izochrony pre výrazne väčšie časy, presnejšie pre 1, 2, 3 a 4 hodiny. Z ilustratívnych dôvodov sme skonštruovali izochrony aj pre 1, 20 a 40 minút. Na Obrázku 4.6 sú znázornené všetky izochrony. Význam červených a zelených bodov je rovnaký ako v oddieloch 4.2.1 a 4.2.2.



Obr. 4.6: Izochrony, 130-ciferné čísla

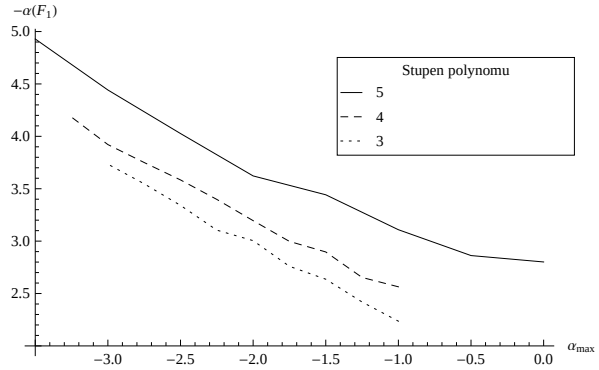
S výnimkou izochrony pre čas jedna minúta sa situácia výrazne zmenila. Najlepšie polynómy boli nájdené pri voľbe $\alpha_{max} = -3$, resp. pri $\alpha_{max} = -2,5$ pre čas dve hodiny a najhoršie pri voľbe vysokého α_{max}. Rozdiel medzi polynómom s najvyšším a najnižším ohodnotením nie je až taký veľký. Priemerná hodnota $\mathbb{E}(F_1)$ v najlepšom bode izochrony v porovnaní s priemernou hodnotou $\mathbb{E}(F_1)$ v najhoršom bode je na každej izochrone len o 7 % vyššia.

Pri generovaní polynómov piateho stupňa sme opäť museli voliť väčšie χ v porovnaní s polynómami nižšieho stupňa. Pre porovnateľné časy (1, 20 a 40 minút) bola voľba χ pri nastavení $\alpha_{max} = -1$ tri až päťkrát väčšie oproti nastaveniu s rovnakým α_{max} pri polynómoch štvrtého stupňa (pri 80-ciferných číslach). Pre $\alpha_{max} = -3$ stúplo χ približne trojnásobne. Z tohto hľadiska parameter χ opäť nestúpol rovnomerne pre všetky α_{max} a na izochronách sa pomer najväčšieho a najmenšieho χ zmenšil. Na každej zo skonštruovaných izochrón[3] je χ pri voľbe $\alpha_{max} = -3$ približne 7-krát väčšie ako pri $\alpha_{max} = 0$.

Na druhej strane, rozdiel v koreňových vlastnostiach polynómov získaných pre rôzne α_{max} sa takmer vôbec nezmenil. Totiž, pre pevné α_{max} sa hľadanie polynómov

[3] s výnimkou izochrony pre čas 1 minúta, ktorá neobsahuje bod s $\alpha_{max} = -3$

s $\alpha(F_1) < \alpha_{max}$ pomocou Algoritmu 3.2 neskomplikuje pri zvýšení stupňa generovaných polynómov. Dokonca môžeme očakávať mierne zrýchlenie, keďže pre polynómy vyššieho stupňa môže α nadobúdať nižšie hodnoty. Poznamenajme, že priemerné hodnoty α klesli v porovnaní s polynómami štvrtého stupňa o šesť desatín. Obrázok 4.7 znázorňuje závislosť $\alpha(F_1)$ na α_{max} pre polynómy stupňa 5 a pre ilustráciu aj pre polynómy tretieho a štvrtého stupňa.

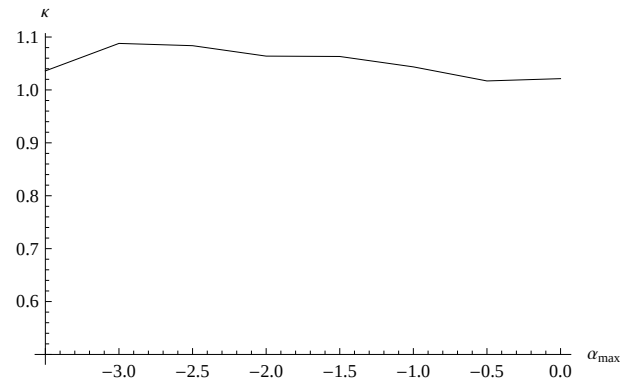


Obr. 4.7: α_{max} versus α, rôzne stupne polynómov

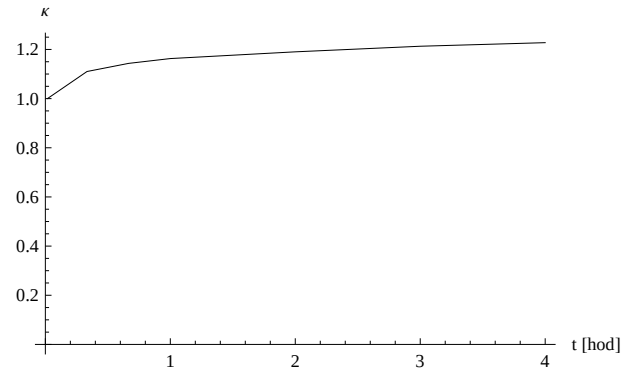
Zmena v rozložení najlepších a najhorších bodov izochrony je jednak spôsobená zmenou vo voľbe χ. Zároveň je zapríčinená faktom, že náročnosť hľadania polynómov s nadpriemernými koreňovými vlastnosťami (v Algoritme 3.2) nerastie so zvyšujúcim sa stupňom generovaných polynómov. Pre polynómy piateho stupňa sa už oplatí viac zamerať na koreňové vlastnosti. To ale neznamená, že máme voliť parameter α_{max} najmenší možný. Pri voľbe $\alpha_{max} = -3,5$ sa kvalita polynómov zhoršila v porovnaní s $\alpha_{max} = -3$. Dôvodom je, že pri takom nastavení sme museli voliť χ približne 2-krát väčšie ako pre $\alpha_{max} = -3$ (pri zachovaní doby výpočtu).

Obrázok 4.8 znázorňuje ohodnotenie polynómu získaného pre rôzne α_{max} na izochrone pre čas tri hodiny. Hodnota κ (zvislá os) je podielom $\mathbb{E}(F_1)$ pre dané α_{max} a najnižšieho ohodnotenia získaného na tejto izochrone. $\mathbb{E}(F_1)$ sa správa podobne aj na ostatných izochronách. Ako vidíme, najkvalitnejšie polynómy sme získali pri nastaveniach $\alpha_{max} = -3$ a $-2,5$. Rozdiel medzi nimi je minimálny.

Z experimentov vyplýva, že α_{max} by sme mali vyberať z intervalu $[-3; -2,5]$ a k nemu voliť vhodné χ, aby výpočet trval nami požadovanú dobu. Na Obrázku 4.9 sú zakreslené najlepšie ohodnotenia získané pre skonštruované izochrony. Tieto sú porovnané vzhľadom k najvyššiemu ohodnoteniu získanému na izochrone pre čas 1 minúta. Obrázok 4.9 teda ukazuje, akú zmenu v kvalite polynómov môžeme očakávať pri optimálnom nastavení α_{max}, χ pre daný čas výpočtu oproti času 1 minúta.



Obr. 4.8: Relatívny nárast $\mathbb{E}(F_1)$ na izochrone pre čas 3 hodiny



Obr. 4.9: Relatívny nárast $\mathbb{E}(F_1)$ pri optimálnom nastavení α_{max} a χ, 130-ciferné čísla

Tabuľka 4.3 obsahuje „najlepšie“ body vybraných izochrón a relatívny nárast hodnoty $\mathbb{E}(F_1)$ voči najvyššiemu ohodnoteniu získanému na izochrone pre čas jedna minúta.

4.2.4 Parameter K

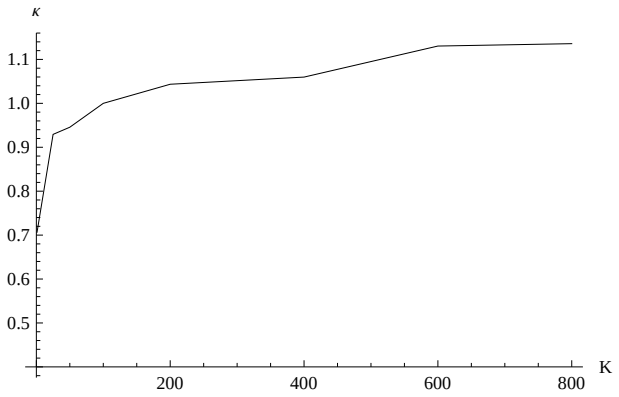
V tomto oddiele budeme skúmať závislosť výpočtového času a kvality výsledných polynómov na počte kandidátov postupujúcich do druhej fázy (parameter K) Algoritmu 3.2 pri konštantnom nastavení α_{max} a χ.

Pri experimente vykonanom pre desať 80-ciferných čísel sme konštruovali polynómy štvrtého stupňa. Algoritmus 3.2 sme spustili s parametrami $\alpha_{max} = -1,5$ a $\chi = 1/90$. Parameter K sme vyberali z intervalu [1,800].

čas (hod)	α_{max}	$1/\chi$	κ
1/60	-1,5	32	1,00
1/3	-3,0	15	1,11
1	-3,0	20	1,16
4	-3,0	28	1,23

Tabuľka 4.3: Optimálne nastavenie parametrov α_{max}, χ a relatívny nárast $\mathbb{E}(F_1)$ pre vybrané časy, 130-ciferné čísla

Podľa očakávaní, so zvyšujúcim sa K doba výpočtu rastie lineárne (v závislosti na K). Kvalita výsledku tiež pri väčšom K vzrástla. Na Obrázku 4.10 je znázornený relatívny nárast $\mathbb{E}(F_1)$ pri vyššie spomínanom experimente. Hodnota κ je podielom priemerného ohodnotenia polynómov získaných pre aktuálne K a priemerného ohodnotenia polynómov získaných pre $K = 100$.

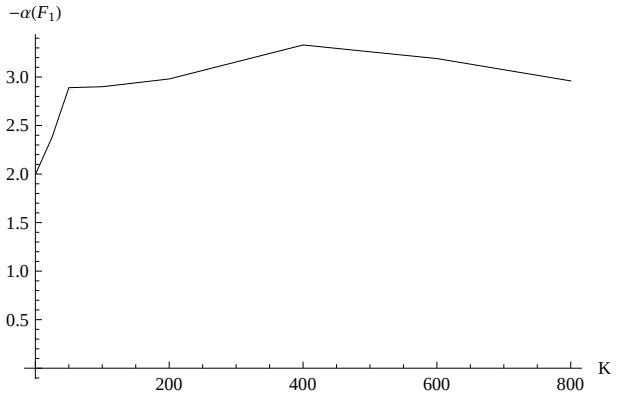


Obr. 4.10: Relatívny nárast $\mathbb{E}(F_1)$ pri náraste K

Dôvodom nárastu je fakt, že pri zvýšení K prehľadáme väčší priestor polynómov a v druhej fáze algoritmu vyberáme výsledný polynóm z väčšej množiny. Tým pádom sa zvýši šanca, že narazíme na kvalitnejšie polynómy.

Na Obrázku 4.11 sú znázornené priemerné hodnoty $\alpha(F_1)$ výsledných polynómov. Ako vidíme, priemerná hodnota $\alpha(F_1)$ klesá až po $K = 400$. Pre $K = 600$ a 800 už hodnota stúpla. Rozdiel medzi $\alpha(F_1)$ pri $K = 800$ a $K = 100$ je približne iba šesť

stotín, napriek tomu kvalita polynómov v porovnaní s $K = 100$ vzrástla až o 14 %. To poukazuje na fakt, že pri zvýšení K nájdeme aj polynómy s menšími koeficientami a že pri výbere v druhej fáze algoritmu pre $K = 600$ a 800 bol v priemernom prípade uprednostnený síce polynóm s horšími koreňovými vlastnosťami, ale s menšími koeficientami.



Obr. 4.11: α versus K

Kapitola 5

Záver

Pri experimentoch s generovaním neskosených polynómov (Algoritmus 3.2) sme skúmali výsledky pre rôzne nastavenia parametrov algoritmu. Najskôr sme skúmali závislosť kvality výsledku na parametroch α_{max} a χ pre pevne nastavené $K = 100$. Potom sme sa zamerali na zmenu v kvalite výsledných polynómov pri meniacom sa K a pevne zvolenom α_{max}, χ.

Optimálne nastavenie α_{max} a χ.

Pre polynomiálne páry stupňa (3,1) sa ukázala ako najlepšia kombinácia voľba najmenšieho možného χ a konštruované polynómy vôbec neselektovať na základe ich koreňových vlastností (to odpovedá nastaveniu parametru $\alpha_{max} = 0$).

Situácia sa mierne zmenila pre polynomiálne páry stupňa (4,1). Nájdenie χ-malého polynómu (pre pevne zvolené χ) je už náročnejšie. Dôsledkom toho je, že si nemôžeme dovoliť extrémne nízke nastavenie parametru χ ako v prípade polynomiálnych párov stupňa (3,1). Na druhej strane, hľadanie polynómov s kvalitnými koreňovými vlastnosťami sa zvýšením stupňa polynómov nijak neskomplikovalo. Optimálne nastavenie parametrov už nie je také jednoznačné. Najlepšie výsledky sme dosahovali pri voľbe α_{max} z intervalu $[-2, -1]$ a k nemu príslušnému χ (tak, aby výpočet trval nami požadovanú dobu).

Pri zvýšení stupňa polynomiálnych párov na (5,1) sa už viac oplatilo zamerať sa na hľadanie polynómov s výnimočne dobrými koreňovými vlastnosťami. Hľadanie χ-malých polynómov sa opäť skomplikovalo, kvôli čomu sme znovu museli voliť vyššie nastavenia parametru χ. Optimálnou voľbou sa tentokrát ukázalo voliť α_{max} z intervalu $[-3; -2, 5]$ a k nemu príslušné χ.

Parameter K

Pri experimentoch s meniacim sa parametrom K sa ukázalo, že výpočtová doba závisí na K lineárne. Kvalita výsledku tiež rastie, ale oveľa pomalšie. Pri osemnásobnom zvýšení z $K = 100$ na $K = 800$ vzrástla v priemere len o 14 %.

Nedostatky

Pre generovanie polynómov v prvej fáze NFS existuje algoritmus na konštrukciu kvalitnejších polynómov, tzv. skosených polynómov (spomenuté v Oddiele 3.3). Výsledky a doba výpočtu tohto algoritmu taktiež závisia od nastavení jeho vnútorných parametrov. Na hľadanie optimálnych nastavení pre tento algoritmus nám nezvýšil čas.

Literatúra

[1] Bach E., Peralta R.: *Asymptotic Semismoothness Probabilities*, Math. Comp. **65** (1996), 1717–1735.

[2] Buhler J. P., Lenstra H. W. Jr, Pomerance C.: „Factoring Integers with the Number Field Sieve“, *The Development of the Number Field Sieve*, LNM **1554** (1993), 50–94.

[3] Knuth D. E., Pardo L. T.: *Analysis of a Simple Factorization Algorithm*, Theor. Comp. Sci. **3** (1976), 321–348.

[4] Lenstra A. K., Lenstra H. W. Jr, Pollard J. M.: *The Development of the Number Field Sieve*, LNM **1554** (1993)

[5] Murphy B. A.: *Polynomial Selection for the Number Field Sieve Integer Factorisation Algorithm*, Dizertačná práca, The Australian National University (1999).

[6] Norton K. K., *Numbers with small prime factors, and the least k-th power nonresidue*, Memoirs Amer. Math. Soc. **106** (1971), 1–106.

[7] Yan S. Y.: *Primality testing and integer factorization in public-key cryptography*, Kluwer Academic Publishers (2004), 139–167.

[8] Wikipedia: *The Magic Words are Squeamish Ossifrage*, http://en.wikipedia.org/wiki/The_Magic_Words_are_Squeamish_Ossifrage

[9] Wikipedia: *RSA Factoring Challenge*, http://en.wikipedia.org/wiki/RSA_Factoring_Challenge