MATHEMATICS OF COMPUTATION
Volume 68, Number 226, April 1999, Pages 859–867
S 0025-5718(99)01003-0

APPLYING SIEVING TO THE COMPUTATION OF QUADRATIC CLASS GROUPS

MICHAEL J. JACOBSON, JR.

Abstract. We present a new algorithm for computing the ideal class group of an imaginary quadratic order which is based on the multiple polynomial version of the quadratic sieve factoring algorithm. Although no formal analysis is given, we conjecture that our algorithm has sub-exponential complexity, and computational experience shows that it is significantly faster in practice than existing algorithms.

1. Introduction

Let $\mathcal{O}_\Delta$, $\Delta < 0$, be the imaginary quadratic order of discriminant Δ. By Cl we denote the class group of $\mathcal{O}_\Delta$, the factor group of all invertible $\mathcal{O}_\Delta$-ideals divided by the subgroup of principal fractional $\mathcal{O}_\Delta$-ideals, and by $h = |Cl|$ we denote the class number. Equivalently, we can consider Cl as the group of equivalence classes of positive definite binary quadratic forms of discriminant Δ with respect to $GL(2, \mathbb{Z})$. We will freely interchange between the two models by means of the map

$$\phi : f \to \mathfrak{a},$$
$$aX^2 + bXY + cY^2 \to a\mathbb{Z} + \frac{b+\sqrt{\Delta}}{2}\mathbb{Z},$$

which converts a binary quadratic form of discriminant Δ to an ideal of $\mathcal{O}_\Delta$ and preserves the isomorphism between both definitions of the class group, i.e., $f \sim g$ if and only if $\phi(f) \sim \phi(g)$.

A well-known problem in computational number theory is the computation of the structure of the class group. More precisely, one wishes to compute the *elementary divisors* of Cl, i.e., the positive integers $m_1, \dots, m_s$ such that $m_i \mid m_{i+1}$ for $1 \le i < s$ and

$$Cl \cong \bigoplus_{i=1}^{s} \mathbb{Z}/m_i\mathbb{Z}.$$

Currently, the best available algorithm is due to Hafner and McCurley [11], and has expected running time $L_\Delta(\sqrt{2})$ under the assumption of the Extended Riemann Hypothesis (ERH), where

$$L_\Delta(\beta) = \exp\left(\sqrt{\log|\Delta| \log\log|\Delta|}\right)^{\beta+o(1)}.$$

Received by the editor May 5, 1997 and, in revised form, August 4, 1997.
1991 *Mathematics Subject Classification.* Primary 11Y40; Secondary 11Y16.
Key words and phrases. Quadratic order, class group, sieving.
The author is supported by the Natural Sciences and Engineering Research Council of Canada.

Their algorithm is based on a factoring algorithm due to Seysen [14], in which relations are generated by factoring random forms using trial division.

Since Hafner and McCurley's algorithm appeared, factoring algorithms have greatly improved in efficiency. Most of the improvements are due to replacing trial division by sieving techniques. These new algorithms not only have better conjectural complexities than previous algorithms, but have also enabled integers of well over 100 decimal digits to be factored, a huge increase over the size of about 60 digits that was previously manageable.

In contrast, class group computation has lagged behind. Buchmann and Düllman [7, 6] have proposed and implemented a more practical version of Hafner and McCurley's algorithm. The largest discriminant for which the class group has been computed with this algorithm has 55 decimal digits. Its class group was computed by Düllman [6] in about 10 days on a distributed system of workstations using trial division combined with the single large prime variant.

In this paper, we present a new algorithm for computing class groups of imaginary quadratic orders. Like Hafner and McCurley's algorithm, our algorithm is also based on a modern factoring algorithm, namely the multiple polynomial version of the quadratic sieve (MPQS) [15]. In the next two sections, we describe our algorithm and present some computational results we have obtained using it. Finally, we discuss some possible enhancements to our algorithm.

2. The algorithm

Our algorithm follows the same general strategy as that in [11]. We first compute a factor base FB consisting of invertible prime ideals such that some subset of FB generates Cl. If $\left(\frac{\Delta}{p}\right) \in \{0,1\}$ for some prime p (where $\left(\frac{x}{y}\right)$ denotes the Kronecker symbol), then the prime ideal corresponding to p is given by

$$\mathfrak{p} = \mathfrak{p}(p) = p\mathbb{Z} + \frac{b_p + \sqrt{\Delta}}{2}\mathbb{Z}, \tag{1}$$

where $0 \leq b_p \leq p$ and $b_p^2 \equiv \Delta \pmod{4p}$. The following well-known theorem allows one to factor any given ideal into a distinct power product of prime ideals.

Theorem 2.1. *If for some invertible ideal $\mathfrak{a} = a\mathbb{Z} + \frac{b+\sqrt{\Delta}}{2}\mathbb{Z}$ we have*

$$N(\mathfrak{a}) = \prod_{p \ prime} p^{t(p)},$$

where by $N(\mathfrak{a})$ $(= a)$ we denote the norm of $\mathfrak{a}$, then $\mathfrak{a}$ is equivalent to

$$\prod_{p \ prime} \mathfrak{p}(p)^{e(p)t(p)},$$

where $e(p) \in \{-1, 1\}$ is such that $b \equiv e(p)b_p \pmod{2p}$.

Let $k = |FB|$. For $\vec{v} \in \mathbb{Z}^k$, $\vec{v} = (v_1, \ldots, v_k)^T$, we define

$$FB^{\vec{v}} = \prod_{i=1}^{k} \mathfrak{p}_i^{v_i}, \tag{2}$$

where $\mathfrak{p}_i \in FB$. We call $\vec{v}$ a *relation* if $FB^{\vec{v}} \sim \mathcal{O}_\Delta$, i.e., the ideal given by $FB^{\vec{v}}$ is principal. Our algorithm produces a generating system $A = (\vec{v}_1, \ldots, \vec{v}_l)$ of the *relation lattice*

$$\Lambda = \{\vec{v} \in \mathbb{Z}^k \mid FB^{\vec{v}} \sim \mathcal{O}_\Delta\}. \tag{3}$$

That relation lattice is the kernel of the homomorphism

$$\mathbb{Z}^k \to Cl, \vec{v} \to FB^{\vec{v}}. \tag{4}$$

Since the classes of the ideals of FB generate the class group, it follows that the homomorphism (4) is surjective, and we have

$$Cl \cong \mathbb{Z}^k/\Lambda.$$

This implies that Λ is a k-dimensional lattice and its determinant is equal to h. The diagonal elements which are greater than 1 in S, the Smith normal form of A, are precisely the elementary divisors of Cl. If $S = UAV$, with $U, V \in GL(k, \mathbb{Z})$, then at very little extra cost one can compute a system of generators of Cl using the transformation matrix U (for details see [10]).

The major difference between our approach and that of [11] is in the way the generating system A of Λ is produced. Hafner and McCurley's solution was to attempt to factor randomly produced ideals over the factor base. We replace this step by a sieve-based strategy similar to that used in the MPQS factoring algorithm.

We need one important observation in order to apply the MPQS to class group computation. This observation is a well-known property of binary quadratic forms, and was pointed out by Seysen [14] as a possible improvement of his factoring algorithm. It is also used by Paulus [13] in his more general algorithm for computing the ideal class group of a quadratic extension over a Euclidean ring.

Proposition 2.2. *If a form $f = aX^2 + bXY + cY^2$ represents an integer n, i.e., $f(x, y) = n$ for some $x, y \in \mathbb{Z}$, then there exists a form $g = nX^2 + b'XY + c'Y^2$ that is equivalent to f.*

Proof. Such a form g can be constructed by solving the linear Diophantine equation

$$ux + vy = 1 \tag{5}$$

for u and v and then applying the transformation matrix

$$\begin{bmatrix} x & -v \\ y & u \end{bmatrix} \in GL(2, \mathbb{Z})$$

to f, yielding the form $g = f(x, y)X^2 + (2(asu + ctv) + b(sv + tu))XY + f(u, v)Y^2 = nX^2 + b'XY + c'Y^2$. Since g is obtained from f via a unimodular transformation of variables, we have $f \sim g$. Note that each solution of (5) results in a different form g, all of which are equivalent to f and have leading coefficient n. □

In the MPQS, one sieves over quadratic polynomials $F(X) = aX^2 + bX + c$ in order to find values of x for which $F(x)$ completely factors over a finite factor base of prime integers. In our case, we compute an ideal $\mathfrak{a}$ as a power product of the prime ideals in our factor base FB and sieve over the corresponding form $f = \phi^{-1}(\mathfrak{a})$ in order to find values of x such that $f(x, 1) = n$ completely factors over the norms of the prime ideals in our factor base. For each such value x, we compute a form $g = nX^2 + b'XY + c'Y^2 \sim f$ using Proposition 2.2 and the ideal $\mathfrak{b} = \phi(g)$. Since we know the factorization of $N(\mathfrak{b}) = n$, we can easily compute the factorization of $\mathfrak{b}$ into prime ideals using Theorem 2.1. Finally, since we also know that $\mathfrak{a} \sim \mathfrak{b}$ (because $f \sim g$), $\mathfrak{b}\mathfrak{a}^{-1}$ is principal, and if $\mathfrak{a} = FB^{\vec{e}}$ and $\mathfrak{b} = FB^{\vec{v}}$ for $\vec{e}, \vec{v} \in \mathbb{Z}^k$, then the vector $\vec{v} - \vec{e}$ is a relation.

We now describe our algorithm in detail. We compute a factor base FB consisting of the k invertible prime ideals of smallest norm. Let $\mathfrak{p}_{\max} = \mathfrak{p}_k$ be the ideal in FB with the largest norm, and let $p_{\max} = N(\mathfrak{p}_{\max})$. A theorem of Bach [2] tells us that in order to ensure that our factor base contains a generating system of Cl (assuming ERH) we need $p_{\max} > 6\log^2|\Delta|$ if Δ is fundamental and $p_{\max} > 12\log^2|\Delta|$ otherwise. Since the linear algebra step in our algorithm is rather expensive, we allow the possibility of using a smaller factor base. In this case, we have to test whether the prime ideals not in the factor base with norms less than $6\log^2|\Delta|$ ($12\log^2|\Delta|$) are contained in the group generated by the prime ideals in the factor base. To check this, as pointed out in [8], it is sufficient to find a principal ideal of the form

$$\mathfrak{p}(p)^{\pm 1} FB^{\vec{v}}, \tag{6}$$

$\vec{v} \in \mathbb{Z}^k$, for each prime p such that $p_{\max} < p \leq 6\log^2|\Delta|$ ($12\log^2|\Delta|$) and $\left(\frac{\Delta}{p}\right) \in \{0, 1\}$. For each such p we compute

$$\mathfrak{a} = \mathfrak{p}(p)^{e_0} FB^{\vec{e}}, \tag{7}$$

where $e_0 = \pm 1$ and the exponent vector $\vec{e} \in \{-1, 0, 1\}^k$ is selected such that

$$N(\mathfrak{a}) \approx \frac{\sqrt{|\Delta|}}{M},$$

where M is the radius of our sieving interval, i.e., each application of sieving will be performed over the interval $(-M, M)$. We set $f = \phi^{-1}(\mathfrak{a})$ and sieve $f(X, 1)$ over $(-M, M)$ using the norms of the prime ideals in FB as potential divisors. If we find some $x \in (-M, M)$ such that $f(x, 1)$ factors completely over the norms of the ideals in FB, we know that some ideal of the form (6) exists and $\mathfrak{p}$ is generated by the ideals in FB. Otherwise, we select another ideal $\mathfrak{a}$ and try again. If after several attempts we are still unable to show that $\mathfrak{p}(p)$ is generated by FB, we add this ideal to FB.

At the moment, we select k from a precomputed list of values which were found to be optimal for the MPQS factoring algorithm implemented in the LiDIA system [4], and the sieve radius is computed using the rule of thumb $M = 4 \times p_{\max}$. Experimental results [12] seem to indicate that the prime ideals of norm less than $12\log|\Delta|$ generate the class group in most cases, and that on average $0.7\log|\Delta|$ is a sufficient bound, so if $p_{\max} \approx \log|\Delta|$ we will probably have a complete generating system.

During the generation of the factor base, we also compute a value h^* such that the class number is guaranteed (again assuming ERH) to lie in the interval $(\frac{h^*}{2}, h^*)$. We first compute $\overline{L(1, \chi_\Delta)}$, an estimate of $L(1, \chi_\Delta)$, via an average of truncated Euler products (see Bach [3]), such that

$$|L(1, \chi_\Delta) - \overline{L(1, \chi_\Delta)}| < \sqrt{2}.$$

Then, from the analytic class number formula,

$$h^* = \frac{\sqrt{2\,|\Delta|}}{\pi} \overline{L(1, \chi_\Delta)} \tag{8}$$

serves our purpose (see [10] for a proof).

At this point, our algorithm diverges somewhat from the algorithm in [11] and more closely follows the MPQS. Again, we set M to be the radius of our sieving interval. We select some value of $k_0 < k$ and compute an ideal $\mathfrak{a}$ such that

$$\mathfrak{a} = FB^{\vec{e}} \tag{9}$$

where the exponent vector $\vec{e} \in \{-1, 0, 1\}^k$ is selected such that $e_i = 0$ for $k_0 < i \leq k$ and

$$N(\mathfrak{a}) \approx \frac{\sqrt{|\Delta|}}{M}.$$

This is similar to the self-initializing variant of the MPQS, where one computes a polynomial whose leading coefficient is of the same size as $N(\mathfrak{a})$ and is also a square-free product of small primes. In the same way as in Buchmann and Düllman's algorithm [7, 6], selecting $k_0 < k$ encourages the relation matrix to be sparse for the rows k_0 to k (see [7, 6]). We select $k_0 = k/50$, a somewhat larger value than that suggested in [7, 6], since our exponents are selected from a smaller set and we want to ensure that there are sufficiently many possible exponent vectors to choose from. Now set $f = \phi^{-1}(\mathfrak{a}) = aX^2 + bXY + cY^2$, $F(X) = f(X, 1) = aX^2 + bX + c$, and sieve $F(X)$ over the interval $(-M, M)$ using the norms of the prime ideals in FB as potential divisors. If for some $x \in (-M, M)$ and integers $\overline{w}_i$, $1 \leq i \leq k$, we have

$$F(x) = \prod_{i=1}^{k} N(\mathfrak{p}_i)^{\overline{w}_i} = n,$$

then f represents n at $(x, 1)$ and we can apply Proposition 2.2 to compute a form $g = nX^2 + b'XY + c'Y^2 \sim f$ and an ideal $\mathfrak{b} = \phi(g)$ with $N(\mathfrak{b}) = n$. Since we know the factorization of $N(\mathfrak{b})$, we can apply Theorem 2.1 to compute exponents $w_i = \pm\overline{w}_i$ such that

$$\mathfrak{b} = FB^{\vec{w}} \tag{10}$$

is the complete factorization of $\mathfrak{b}$ over FB. Finally, since $\mathfrak{a} \sim \mathfrak{b}$, we have

$$\mathfrak{b}\mathfrak{a}^{-1} = FB^{\vec{w}} FB^{-\vec{e}} = FB^{\vec{w}-\vec{e}} \sim \mathcal{O}_\Delta,$$

and the vector $\vec{v} = \vec{w} - \vec{e}$ is a relation. We add this relation to the relation matrix $\overline{A}$, our potential generating system of the relation lattice Λ.

We continue to produce relations until we have at least $|FB| + c$ of them for some small constant c (we found $c = 20$ was normally sufficient). Since we want the relation matrix to be non-singular, it is necessary that each prime ideal in FB be represented in at least one relation. This is by no means a guarantee that the matrix will be non-singular, but it seems to work well in practice. For each $\mathfrak{p}_j \in FB$ such that $v_j = 0$ for all relations $\vec{v} \in \overline{A}$, we compute an ideal $\mathfrak{a}$ as in (9), except here we force $e_j = \pm 1$. We also select e_i from $\{-1, 0, 1\}$ for all $i \leq k$, $i \neq j$, since usually only a small number of relations are generated in this step and the overall sparseness of the matrix is not greatly affected. We execute the sieving step on $\mathfrak{a}$ as before and add to $\overline{A}$ the first relation $\vec{v}$ that we find which has $v_j \neq 0$. We repeat this step for each such $\mathfrak{p}_j$ until every ideal in FB is represented in a relation.

The rest of the algorithm follows very closely that of [11]. We compute the Hermite normal form of the relation matrix $\overline{A}$, and if $\det \overline{A} = 0$ or $\det \overline{A} > h^*$, we compute a few more relations (we used 20 here). When $0 < \det \overline{A} < h^*$, we know that $A = \overline{A}$ is a complete generating system of Λ. We compute S, the Smith normal form of A, and the diagonal elements of S which are greater than 1 are the elementary divisors of Cl.

We summarize our method in the following algorithm.

Algorithm 2.1.

Input: $\Delta < 0$ (the discriminant of $\mathcal{O}_\Delta$), M, k_0, and k as described above.
Output: $m_1, \ldots, m_s$, the elementary divisors of Cl.

1. Set $\overline{A} = ()$. Compute h^* from (8). Compute FB as above such that $|FB| = k$, and set $p_{\max} = N(\mathfrak{p}_k)$. If $p_{\max} > 12 \log^2 |\Delta|$ ($6 \log^2 |\Delta|$ if Δ is fundamental), go to Step 3.
2. For each $\mathfrak{p}$ such that $p_{\max} < N(\mathfrak{p}) \leq 12 \log^2 |\Delta|$ ($6 \log^2 |\Delta|$ if Δ is fundamental):
   (a) Compute $\mathfrak{a}$ as in (7).
   (b) Set $f = \phi^{-1}(\mathfrak{a})$ and sieve $f(X, 1)$ over $(-M, M)$. If there is no $x \in (-M, M)$ such that $f(x, 1)$ factors completely over the norms of the ideals in FB, go to Step 2a. If we have tried 1000 different ideals without success, add $\mathfrak{p}$ to FB.
3. Compute $\vec{e} \in \{-1, 0, 1\}^k$ and $\mathfrak{a}$ as in (9).
4. Set $f = \phi^{-1}(\mathfrak{a})$, $F(X) = f(X, 1)$. Sieve $F(X)$ over $(-M, M)$.
5. For each $x \in (-M, M)$ such that $F(x)$ completely factors over the norms of the prime ideals in FB :
   (a) Compute the exponents $\overline{w}_i$ such that $F(x) = \prod_{i=1}^{k} N(\mathfrak{p}_i)^{\overline{w}_i}$.
   (b) Solve $ux + v = 1$.
   (c) Compute g by applying the transformation matrix $\left[\begin{smallmatrix} x & -v \\ 1 & u \end{smallmatrix}\right]$ to f. Compute $\mathfrak{b} = \phi(g)$.
   (d) Compute $\vec{w}$ such that $w_i = \pm \overline{w}_i$ and $\mathfrak{b} = FB^{\vec{w}}$, using Theorem 2.1.
   (e) Set $\overline{A} = (\overline{A}, \vec{v})$, where $\vec{v} = \vec{w} - \vec{e}$.
6. If the number of relations we have computed is less than $|FB| + 20$, go to Step 3.
7. For each $\mathfrak{p}_j \in FB$ such that $v_j = 0$ for all relations $\vec{v} \in \overline{A}$, execute Step 3 to Step 5e. Force $e_j = 1$ in Step 3 and exit Step 5 after the first relation $\vec{v}$ with $v_j \neq 0$ is found.
8. Compute the Hermite normal form of $\overline{A}$ and set $h = \det \overline{A}$. If $h = 0$ or $h > h^*$, execute Step 3 to Step 5e until 20 more relations have been found, and repeat Step 8.
9. Compute S, the Smith normal form of $\overline{A}$, and return the diagonal elements of S that are greater than 1.

As stated above, the parameters M, k_0, and k are selected based on knowledge and experience from other algorithms. It is probable that further experiments will enable us to find optimal values for our algorithm. In particular, since the linear algebra step of our algorithm is somewhat more difficult than that of factoring algorithms, it is likely that a slightly smaller factor base will turn out to be optimal.

If we ensure that the factor base and sieve radius have sub-exponential size, then it is reasonable to expect that we can generate $O(|FB|)$ relations in about the same time as the MPQS, since these parts of the algorithms are so similar. However, since the relations are generated by a process which is not completely random, it is not clear to us how to analyze the probability that a new relation lies outside the current relation lattice. Thus, we are unable to compute the expected number of relations required to generate a relation matrix with full rank, and a complete analysis of our algorithm eludes us.

3. Computational results

We have implemented most of our algorithm as part of the LiDIA system, which is currently being developed at the Technische Hochschule Darmstadt [4]. The Hermite normal form computations were done by Patrick Theobald using special techniques which exploit the sparseness of the relation matrices and are not yet implemented in LiDIA.

We present the results of applying our algorithm to compute the class groups of four imaginary quadratic orders with various size discriminants. Table 1 gives the discriminant, factorization of the discriminant (computed from a system of generators of the class group), class number, and elementary divisors of the class group for each of these orders. The number in parenthesis after the discriminant is the number of decimal digits. The class group is presented as $[m_1\ m_2\ \ldots m_s]$, where the m_i are the elementary divisors.

Table 1. Class groups of some imaginary quadratic orders

Δ_1	$-4 \times F_7 = 4 \times (2^{2^7} + 1)$ (40) $= -1 \times 2^2 \times 59649589127497217 \times 5704689200685129054721$
h	17787144930223461408
Cl	[2 8893572465111730704]
Δ_2	$-(4 \times 10^{54} + 4)$ (55) $= -1 \times 2^2 \times 101 \times 109 \times 9901 \times 153469 \times 999999000001 \times 597795771563/$ 34533866654838281
h	1056175002108254379317829632
Cl	[2 2 2 2 2 33005468815882949353682176]
Δ_3	$-56759029509462061499204078404947821190422701840487390196283$ (59) $= -1 \times 235942923943814840172714410183 \times 2405625418246410575130433/$ 26701
h	34708563502858399116135176220
Cl	[34708563502858399116135176220]
Δ_4	$-(4 \times 10^{64} + 4)$ (65) $= -1 \times 2^2 \times 1265011073 \times 15343168188889137818369 \times 515217525265213/$ 267447869906815873
h	178397819605839608466892693850112
Cl	[4 4 11149863725364975529180793365632]
Δ_5	$-46952046735522451306774137871578512166228058934334430430/$ 26971349460603 (70)
h	???
Cl	???

Table 2 contains some of the run-time statistics collected during the generation of the relation matrices. Here, k, k_0, and M are as defined above, "# rels" is the number of relations computed, "# forms" is the number of forms which were generated in Step 3 of Algorithm 2.1, t_2 is the CPU time in minutes required by Step 2 (if required), t_L is the is the total CPU time in minutes required to generate the relation matrix, i.e., Steps 1 to 6 in Algorithm 2.1, and "time" is the total CPU

TABLE 2. Run-time statistics

Δ	k	k_0	$p_{\max}$	# rels	M	# forms	t_2	t_L	time
Δ_1	1000	20	17389	1058	69556	480	4.15	5.75	7.95
Δ_2	4100	82	83459	4254	333836	5503	12.15	108.20	532.24
Δ_3	5500	110	117577	5685	470308	16255	-	315.75	2338.28
Δ_4	7300	146	157243	7579	628972	27369	-	855.05	6457.28
Δ_5	8800	176	194771	9041	779084	143678	-	5007.42	?

time in minutes required by Algorithm 2.1. The computations were all carried out on a SPARC-ultra computer.

We knew beforehand that all of our discriminants were fundamental, so we were able to use the upper bound $6\log^2|\Delta|$ in Step 2. For discriminants Δ_1 and Δ_2, the factor bases used did not contain all the prime ideals with norms less than this bound, so it was necessary to execute Step 2.

Δ_2 is the 55-digit discriminant for which the class group of its corresponding imaginary quadratic order was computed by Buchmann and Düllman [6]. Not only were we able to compute this class group in a fraction of the time they needed using a single computer without large prime variation, but we were also able to compute class groups for two significantly larger discriminants. In addition, we have computed a relation matrix for Δ_5, a 70-digit discriminant, but so far we have been unable to finish the Hermite normal form computation.

4. CONCLUSION

One obvious improvement to our algorithm is to incorporate a large prime or double large prime strategy in a similar fashion to the MPQS. In factoring algorithms, it is sufficient to find a set of partial relations ($f(x)$ factors completely over the factor base except for one or two slightly larger primes) such that when these partial relations are combined (multiplied together), the exponents of the large primes in the combined relation are all even. In our case, a partial relation consists of a principal ideal which completely factors over our factor base except for one or two prime ideals of slightly larger norms than the prime ideals in the factor base. These large prime ideals may have exponents of $+1$ or -1 (see Theorem 2.1). We want to find a product of principal ideals represented by partial relations (or their inverses) such that the exponents of the large prime ideals actually cancel, and we are left with a principal ideal which completely factors over the factor base. Buchmann and Düllman [6] showed how a single large prime strategy can effectively be used, and it should not be a problem to extend their method to handle two large primes.

Our algorithm should also be very effective in computing class groups of real quadratic orders. For each relation $\vec{v}$ one would also have to compute a minimum α such that $FB^{\vec{v}} \sim (\alpha)$. Then, the methods described in [5], [9], and [1] can be applied directly. Further experiments are currently underway in these directions.

REFERENCES

1. C.S. Abel, *Ein Algorithmus zur Berechnung der Klassenzahl und des Regulators reellquadratischer Ordnungen*, Ph.D. thesis, Universität des Saarlandes, Saarbrücken, Germany, 1994.

2. E. Bach, *Explicit bounds for primality testing and related problems*, Math. Comp. **55** (1990), 355–380. MR **91m:**11096
3. ———, *Improved approximations for Euler products*, Number Theory: CMS Proc., vol. 15, Amer. Math. Soc., Providence, RI, 1995, pp. 13–28. MR **96i:**11124
4. I. Biehl, J. Buchmann, and T. Papanikolaou, *LiDIA - a library for computational number theory*, The LiDIA Group, Universität des Saarlandes, Saarbrücken, Germany, 1995.
5. J. Buchmann, *A subexponential algorithm for the determination of class groups and regulators of algebraic number fields*, Séminaire de Théorie des Nombres (Paris), 1988-89, pp. 27–41. MR **92g:**11125
6. J. Buchmann and S. Düllmann, *Distributed class group computation*, Festschrift aus Anlaß des sechzigsten Geburtstages von Herrn Prof. Dr. G. Hotz, Universität des Saarlandes, 1991, and Teubner, Stuttgart, 1992, pp. 69–79. MR **93e:**11153
7. ———, *A probabilistic class group and regulator algorithm and its implementation*, Computational Number Theory, Walter de Gruyter & Co., New York, 1991, pp. 53–72. MR **92m:**11150
8. H. Cohen, *A course in computational algebraic number theory*, Springer-Verlag, Berlin, 1993. MR **94i:**11105
9. H. Cohen, F. Diaz y Diaz, and M. Olivier, *Calculs de nombres de classes et de régulateurs de corps quadratiques en temps sous-exponentiel*, Séminaire de Théorie des Nombres (Paris), 1993, pp. 35–46. MR **94m:**11151
10. S. Düllmann, *Ein Algorithmus zur Bestimmung der Klassengruppe positiv definiter binärer quadratischer Formen*, Ph.D. thesis, Universität des Saarlandes, Saarbrücken, Germany, 1991.
11. J.L. Hafner and K.S. McCurley, *A rigorous subexponential algorithm for computation of class groups*, J. Amer. Math. Soc. **2** (1989), 837–850. MR **91f:**11090
12. M.J. Jacobson, Jr., *Some experimental results on ideal class groups of quadratic fields*, Unpublished MS., 1997.
13. S. Paulus, *An algorithm of subexponential type computing the class group of quadratic orders over principal ideal domains*, Algorithmic Number Theory (Université Bordeaux I, Talence, France), Lecture Notes in Computer Sci., vol. 1122, Springer-Verlag, Berlin, 1996, pp. 243–257. MR **98e:**11143
14. M. Seysen, *A probabilistic factorization algorithm with quadratic forms of negative discriminant*, Math. Comp. **48** (1987), 757–780. MR **88d:**11129
15. R.D. Silverman, *The multiple polynomial quadratic sieve*, Math. Comp. **48** (1987), 329–339. MR **88c:**11079

Technische Universität Darmstadt, FB Informatik, Institut für theoretische Informatik, Alexanderstr. 10, 64283 Darmstadt, Germany

E-mail address: jacobs@cdc.informatik.tu-darmstadt.de