Solving Subset Sum Problems of Density close to 1 by "randomized" BKZ-reduction

Claus Peter Schnorr and Taras.Shevchenko

Fachbereich Informatik und Mathematik,
Goethe-Universität Frankfurt, PSF 111932,
D-60054 Frankfurt am Main, Germany.
schnorr@cs.uni-frankfurt.de, taras.s.shevchenko@googlemail.com
November 2, 2012

Abstract. Subset sum or Knapsack problems of dimension n are known to be hardest for knapsacks of density close to 1. These problems are NP-hard for arbitrary n. One can solve such problems either by lattice basis reduction or by optimized birthday algorithms. Recently Becker, Coron, Joux [BCJ10] present a birthday algorithm that follows Schroeppel, Shamir [SS81], and Howgrave-Graham, Joux [HJ10]. This algorithm solves 50 random knapsacks of dimension 80 and density close to 1 in roughly 15 hours on a 2.67 GHz PC. We present an optimized lattice basis reduction algorithm that follows Schnorr, Euchner [SE03] using pruning of Schnorr, Hörner [SH95] that solves such random knapsacks of dimension 80 on average in less than a minute, and 50 such problems all together about 9.4 times faster with less space than [BCJ10] on another 2.67 GHz PC.

Keywords. BKZ, block reduction, pruned BKZ, subset sum problem.

Preliminaries on lattices. A basis matrix $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_n] \in \mathbb{R}^{m\times n}$ of n linear independent vectors $\mathbf{b}_1, ..., \mathbf{b}_n \in \mathbb{R}^m$ generates the lattice $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} \,|\, \mathbf{x} \in \mathbb{Z}^n\} \in \mathbb{R}^m$ of dimension n. Lattice reduction algorithms transform a given basis $\mathbf{B}$ into a basis of $\mathcal{L}(\mathbf{B})$ consisting of short vectors. The length of $\mathbf{b} \in \mathbb{R}^m$ is $\|\mathbf{b}\| = (\mathbf{b}^t\mathbf{b})^{1/2}$. $\lambda_1(\mathcal{L}) = \min_{\mathbf{b}\in\mathcal{L}\setminus\mathbf{0}} \|\mathbf{b}\|$ is the minimal length of nonzero $\mathbf{b} \in \mathcal{L}$. The determinant of $\mathcal{L}$ is $\det \mathcal{L} = (\det \mathbf{B}^t\mathbf{B})^{1/2}$. The Hermite bound $\lambda_1(\mathcal{L})^2 \le \gamma_n(\det \mathcal{L})^{2/n}$ holds for all lattices $\mathcal{L}$ of dimension n and the Hermite constant γ_n is minimal for this property.

The LLL-algorithm of H.W. Lenstra Jr., A.K. Lenstra and L. Lovász [LLL82] transforms a given basis $\mathbf{B}$ in polynomial time into a basis $\mathbf{B}'$ of the same lattice such that $\|\mathbf{b}'_1\| \le \alpha^{\frac{n-1}{2}}\lambda_1$, where $\alpha > 4/3$. It is important to minimize the proven bound on $\|\mathbf{b}_1\|/\lambda_1$ for polynomial time reduction algorithms and to optimize the polynomial time.

The best known algorithms perform blockwise basis reduction for blocksize $k \ge 2$ generalizing the blocksize 2 of LLL-reduction. Schnorr [S87] introduced blockwise HKZ-reduction. [SE94] reports on solving subset sum problems by BKZ-reduction with block size $k \le 50$ and a particular pruning method that has recently be analyzed by N. Gama, P.Q. Nguyen and O. Regev [GNR10]. We will use BKZ with the pruning of [SH95] and the algorithms of NTL [Sh].

Notation. Let $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_n] \in \mathbb{R}^{m\times n}$ be a basis matrix of rank $n = hk$ and $\mathbf{B} = \mathbf{QR}$ be its QR-decomposition, where $\mathbf{R} = [r_{i,j}]_{1\le i,j\le n} \in \mathbb{R}^{n\times n}$ is upper triangular with positive diagonal entries $r_{i,i} > 0$ and $\mathbf{Q} \in \mathbb{R}^{m\times n}$ is isometric with pairwise orthogonal column vectors of length 1. We denote $\mathrm{GNF}(\mathbf{B}) = \mathbf{R}$. Let $\mathbf{R}_\ell = [r_{i,j}]_{k\ell-k+1\le i,j\le k\ell} \in \mathbb{R}^{k\times k}$ be the submatrix of $\mathbf{R} = [r_{i,j}] \in \mathbb{R}^{n\times n}$ for the ℓ-th block of blocksize $k \ge 2$, let $\mathcal{D}_\ell = (\det \mathbf{R}_\ell)^2$.

LLL-bases. [LLL82] A basis $\mathbf{B} = \mathbf{QR} \in \mathbb{R}^{m\times n}$ is LLL-basis for δ, $\frac{1}{4} < \delta \le 1$, $\alpha = 1/(\delta - 1/4)$ if

- $|r_{i,j}| \le \frac{1}{2} r_{i,i}$ holds for all $j > i$,
- $\delta r_{i,i}^2 \le r_{i,i+1}^2 + r_{i+1,i+1}^2$ holds for $i = 1, ..., n-1$.

An LLL-basis $\mathbf{B}$ for δ satisfies $\|\mathbf{b}^*_\ell\|^2/\|\mathbf{b}^*_{\ell+1}\|^2 \le \alpha$ for all $\ell = 1, ..., n-1$ and

$$\|\mathbf{b}_1\| \le \alpha^{\frac{n-1}{4}}(\det \mathcal{L})^{1/n}, \qquad \|\mathbf{b}_1\| \le \alpha^{\frac{n-1}{2}}\lambda_1.$$

The subset sum or knapsack problem.
GIVEN positive integers $a_1, ..., a_n > 0$ and s such that $\max a_i \le s \le \sum_{i=1}^n a_i$
FIND $x_1, ..., x_n \in \{\pm 1\}$ such that $\sum_{i=1}^n a_i x_i = s$ if such $x_1, ..., x_n$ exist.

In the *inverse* problem s is replaced by $\sum_{i=1}^{n} a_i - s$. The inverse problem is solved by negating any solution $x_1, ..., x_n$ of the original problem into $1 - x_1, ..., 1 - x_n$. Obviously it is sufficient to solve either the original or the inverse problem.
For simplicity we only search for knapsack solutions where n is even and $\sum_{i=1}^{n} x_i = n/2$.
The *density* d of the problem is $d := \frac{n}{\log_2 \max a_i}$.

We solve knapsack problems by BKZ- reduction of the lattice basis $\mathbf{B}$ introduced in [SE91].

$$\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_{n+1}] = \begin{bmatrix} 2 & & & 1 \\ & \ddots & O & \vdots \\ & O & 2 & 1 \\ Na_1 & \cdots & Na_n & Ns \\ 0 & \cdots & 0 & 1 \\ N & \cdots & N & \frac{n}{2}N \end{bmatrix} \in \mathbb{Z}^{(n+3)(n+1)}. \tag{1}$$

The last row of $\mathbf{B}$ helps in finding solutions satisfying $\sum_{i=1}^{n} x_i = n/2$.
We report on experiments for $n = 80$ and random $a_1, ..., a_n \in_R [1, 2^n]$. These subset sum problems have density close to 1. In particular we choose an integer $N > \sqrt{n}$ and $N = 16$ for $n = 80$.

Fact. Let $\mathbf{B}$ be a basis (1). Then every lattice vector $\mathbf{b} = (b_1, ..., b_{n+3})^t \in \mathcal{L}(\mathbf{B})$ that satisfies

$$|b_1| = |b_2| = \cdots = |b_n| = 1, \quad |b_{n+2}| = 1, \quad b_{n+1} = b_{n+3} = 0 \tag{2}$$

yields a knapsack solution $x_1, ..., x_n$, $x_i := |b_i - b_{n+2}|/2$, that satisfies $\sum_{i=1}^{n} x_i = n/2$. Conversely every knapsack solution satisfying $\sum_{i=1}^{n} x_i = n/2$ corresponds to some $\mathbf{b} \in \mathcal{L}(\mathbf{B})$ that satisfies (2).

It has been shown in [CJLOSS92] that the shortest, nonzero vector of the lattice with asis $\mathbf{B}$ of (1) yields a knapsack solution for almost all knapsacks of density less than 0.9408 and for sufficiently large n, either for the original or the inverse problem.

Details of the reduction algorithm. (We optimize the approach of [SE94]. Our algorithm performs BKZ reduction for various blocksizes combined with permutations of the basis vectors)

1. Iteratively BKZ-reduce a permutation of the given basis $\mathbf{B}$ without pruning with blocksizes 2, 4, 8, 16, 32. In each round we first permute the columns of the basis matrix such that the first columns have a nonzero entry in row $n + 2$ (then the subsequent BKZ reduction operates heavily on the last column $\mathbf{b}_{n+1}$ of the input basis $\mathbf{B}$), and thern sort the vectors of the basis according to their length preserving the particular initial columns. Always terminate as soon as a solution has been found.

2. BKZ-reduce the input basis with blocksizes 30, 31, 32,..., 61 independently – not iteratively – with pruning according to [SH95]. Always terminate as soon as a solution has been found. Let the pruning parameter s iteratively circulate with the blocksize k through a couple of nearly optimal values.. The range 30,..., 61 of blocksizes k and the pruning parameters 10, 11, 12 are adapted to the dimension $n = 80$.

Comments. Performing BKZ with iteratively doubling the block size k is particularly efficient. For the primal-dual version of BKZ or slide-reduction it has been proved in [S11] that iterative reduction with block sizes 2, 4,..., 2^ℓ is twice as fast than reduction with block size 2^ℓ.
While unpruned BKZ with block size 32 is still pretty fast this is no more true for larger block sizes. Therefore step **2** prunes BKZ for block sizes $k \geq 30$. It turns out that two rounds of step **2** of distinct block sizes and distinct pruning parameters s have success rates that are – roughly – statistically independent. In this way we randomize BKZ.

Here are the results for 50 random subset sum problems of dimension $n = 80$:

round	block size k	pruning par. s	# successes	time per suc., mm : ss
1.	2	no pruning	0	00 : 00
2.	4	no pruning	0	00 : 00
3.	8	no pruning	0	00 : 00
4.	16	no pruning	0	00 : 00
5.	32	no pruning	8	00 : 29
6.	30	10	3	01 : 07
7.	31	11	2	01 : 09
8.	32	12	4	01 : 15
9.	33	10	2	01 : 17
10.	34	11	3	02 : 01
11.	35	12	4	01 : 36
12.	36	10	2	01 : 35
13.	37	11	3	01 : 32
14.	38	12	5	02 : 08
15.	39	10	1	01 : 43
16.	40	11	4	02 : 21
17.	41	12	1	02 : 30
18.	42	10	0	00 : 00
19.	43	11	0	00 : 00
20.	44	12	3	02 : 35
21.	45	10	0	00 : 00
22.	46	11	2	03 : 48
23.	47	12	0	00 : 00
24.	48	10	0	00 : 00
25.	49	11	0	00 : 00
26.	50	12	0	00 : 00
27.	51	10	0	00 : 00
28.	52	11	0	00 : 00
29.	53	12	0	00 : 00
30.	54	10	0	00 : 00
31.	55	11	1	06 : 20
32.	56	12	1	05 : 21
33.	57	10	0	00 : 00
34.	58	11	0	00 : 00
35.	59	12	0	00 : 00
36.	60	10	0	00 : 00
37.	61	11	1	06 : 12

Comments. BKZ with block size 2 in round **1.** means LLL-reduction with $\delta = 0.99$. Here we perform the slightly stronger reduction of LLL with deep insertion of depth 1 of [SE94]. BKZ is also used with $\delta = 0.99$ so that the initial vector $\mathbf{b}_j$ of the block $\mathbf{b}_j, \cdots, \mathbf{b}_{j+k-1}$ is minimized wihin the block block if this decreases $\|\mathbf{b}_j^*\|^2$ by the factor 0.99.

The run time noted in the last column gives the average total time (over all rounds) for the problems solved in that round. Note that the problems whose solutions have been found in previous rounds have already been terminated previously. The total running time is the scalar product of the last two columns. The total time for all 50 problems is 1 hour 34 minutes and 49 seconds, i.e less than a minute per problem.

The three problems that have been solved in rounds 31, 32, 37 together took about 17 minutes and 53 seconds.

Recall that rounds 1 – 5 operate iteratively on the output basis of the previous round while rounds 6 – 37 operate on the original input basis.

The 25 most difficult of the 50 problems together took 34 minutes and 7 seconds. The 25 most easy problems together took only 10 minutes and 52 seconds.

Possible improvements. The 42 problems that are solved in rounds 6 to 32 distribute to the pruning parameters 10, 11, 12 as 8, 16, 18. Surprisingly the time per solution does not increase much within the pruning parameter 10, 11, 12. This indicates that it the pruning parameters 10, 11, 12 might better be replaced by either 11, 12 or 11, 12, 13.

We used an Intel(R) Xeon(R) CPU X5650 @ 2.67 GHz. [BCJ10] solved 50 random knapsack problems of dimension 80 in 14 hours and 50 minutes and using enormous space on an Intel(R) Core$^{\mathrm{TM}}$ i7 CPU M 620 at 2.67 GHz. This is about 9.5 times our time for 50 random problems.

Independence of the success rates of iteratively composed and non composed rounds. For 20 random subset sum problems of dimension 80 and density near 1 we present in two cases the minimal number of rounds of non composed BKZ with block sizes $k = 30, 31, ..., 62$ until a solution is found. The pruning parameter s circulates through 10, 11, 12 as k increases.

Case 1 For input bases where the iteratively composed BKZ-reduction for block sizes $k = 30, 31, ..., 62$ has found a subset sum solution.

Case 2 For input bases where the iteratively composed BKZ-reduction for block sizes $k = 30, 31, ..., 62$ has found no subset sum solution.

Case 1	number of successful k	# of times	Case 2	number of successful k	# of times
	1	1		4	3
	2	0		5	1
	3	2		6	3
	4	3		7	0
	5	1		8	1
	6	1		9	0
	7	3		10	1

This example indicates that solvability or unsolvability by iteratively composed BKZ-reduction for block sizes $k = 30, 31, ..., 82$ had no direct influence on the number of block sizes 30, 31,...,62 for which BKZ-reduction applied to the unchanged input basis finds a subset sum solution.

References

[BCJ10] *A. Becker, J.-S. Coron and A. Joux*, Improved generic algorithms for hard knapsacks. In Proc. Eurocrypt 2011, LNCS 6632, Springer, pp, 364–385, 2010. Extended version in Cryptology ePrint Archive Report 2011/474, 2011.

[CJLOSS92] *M.J.Coster, A. Joux, B.A. La Macchia, A.M. Odlyzko, C.P. Schnorr and J. Stern*, An improved low density subset sum algorithm. *Computational Complexity* 2, pp. 11–128, 1992.

[GNR10] *N. Gama, P.Q. Nguyen and O. Regev*, Lattice enumeration using extreme pruning. In Proc. Eurocrypt 2010, LNCS 6110, Springer, pp. 257– 278, 2010.

[HJ10] *N, Howgrave-Graham and A, Joux*, New generic algorithms for hard knapsacks. In Proc. Eurocrypt 2010, LNCS 6110, Springer, pp. 235–256, 2010.

[LLL82] *H.W. Lenstra Jr., A.K. Lenstra and L. Lovász*, Factoring polynomials with rational coefficients, Mathematische Annalen 261, pp. 515–534, 1982.

[S87] *C.P. Schnorr*, A hierarchy of polynomial time lattice basis reduction algorithms. *Theoret. Comput. Sci.*, **53**, pp. 201–224, 1987.

[S11] *C.P. Schnorr*, Accelerated and improved slide- and LLL-reduction. Electronic Colloquium on Computational Complexity, TR11-050, 2011.

[SE91] *C.P. Schnorr and M. Euchner*, Lattice basis reduction: Improved algorithms and solving subset sum problems. In Proc. of Fundamentals of Computation Theory, FCT'91, LNCS 529, Springer, pp. 68-85, 1991.

[SE94] *C.P. Schnorr and M. Euchner*, Lattce basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical Programming **66**, pp. 181–199, 1994.

[SH95] *C.P. Schnorr and H.H. Hörner*, Attacking the Chor-Rivest cryptosystem by improved lattice reduction, In Proc. Eurocrypt 1995, LNCS 921, Springer, pp. 1–12, 1995.

[SS81] *R. Schroeppel and A. Shamir*, A $T = O(2^{n/2}), S = O(2^{n/4})$ algorithm for certain NP-complete problems. *SIAM J. Compu.*, 10(3), pp. 456–464, 1981.

[Sh] *V. Shoup*, **A Library for doing Number Theory. version 5.5.2.**