Closest Vector Problem

Daniele Micciancio, University of California, San Diego
daniele@cs.ucsd.edu

Revised August 11, 2003

The Closest Vector Problem (CVP) is a computational problem on lattices closely related to SVP. (See Shortest Vector Problem.) Given a lattice $\mathcal{L}$ and a target point $\vec{x}$, CVP asks to find the lattice point closest to the target. As for SVP, CVP can be defined with respect to any norm, but the Euclidean norm is the most common (see the entry lattice for a definition). A more relaxed version of the problem (used mostly in complexity theory) only asks to compute the distance of the target from the lattice, without actually finding the closest lattice vector.

CVP has been studied in mathematics (in the equivalent language of quadratic forms) since the 19th century. One of the first references to CVP (under the name "Nearest Vector Problem") in the computer science literature is [11], where the problem is shown to be NP-hard to solve exactly.

Many applications of the CVP only require finding a lattice vector that is not too far from the target, even if not necessarily the closest. A g-approximation algorithm for CVP finds a lattice vector within distance at most g times the distance of the optimal solution. The best known polynomial time algorithms to solve CVP due to Babai and Kannan [2, 7] are based on lattice reduction, and achieve approximation factors that (in the worst case) are essentially exponential in the dimension of the lattice. In practice, heuristics approaches (e.g., the "embedding technique", see lattice reduction) seem to find relatively good approximations to CVP in a reasonable amount of time when the dimension of the lattice is sufficiently small.

CVP is widely regarded, both in theory and in practice, as a considerably harder problem than SVP. CVP is known to be NP-hard to solve approximately within any constant factor or even some slowly increasing (sub-polynomial) function of the dimension n [1, 3]. However, CVP is unlikely to be NP-hard to approximate within small polynomial factors $g = O(\sqrt{n/\log n})$ [5]. Goldreich, Micciancio, Safra and Seifert [6] showed that any algorithm to efficiently approximate CVP can be used to efficiently approximate SVP within the same approximation factor and with essentially the same computational effort, formalizing the intuition that CVP is not an easier (and is a possibly harder) problem than SVP.

CVP is the basis of various cryptosystems (see lattice based cryptography) where the decryption process corresponds roughly to a CVP computation. These cryptosystems are based on the fact that any lattice admits many different representations, (e.g., it can be represented by different bases,) and some of them may have better geometric properties than others, so that they can be used as a decryption trapdoor. However, there are lattices [8, 4, 10] that admit no good representation, i.e., solving CVP (even approximately) is NP-hard no matter which basis (or other auxiliary information) is given. Therefore, the CVP instances used by lattice based cryptosystems (for which CVP can be efficiently solved using the decryption key) are conceivably easier than general CVP instances.

CVP has many applications in computer science, beside cryptography, and finding a good CVP approximation algorithm with approximation factors that grow as a polynomial in the dimension of the lattice, is one of the major open problems in the area. For further information about CVP and other computational problems on lattices the reader is referred to the book [9].

References

[1] S. Arora, L. Babai, J. Stern, and E. Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, Apr. 1997. Preliminary version in FOCS'93.

[2] L. Babai. On Lovasz' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[3] I. Dinur, G. Kindler, and S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. In *39th Annual Symposium on Foundations of Computer Science*, pages 99–111, Palo Alto, California, Nov. 1998. IEEE.

[4] U. Feige and D. Micciancio. The inapproximability of lattice and coding problems with preprocessing. *Journal of Computer and System Sciences*, 2003. To appear. Preliminary version in CCC 2002.

[5] O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *Journal of Computer and System Sciences*, 60(3):540–563, 2000. Preliminary version in STOC'98.

[6] O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.

[7] R. Kannan. *Annual reviews of computer science*, volume 2, chapter Algorithmic Geometry of numbers, pages 231–267. Annual Review Inc., Palo Alto, California, 1987.

[8] D. Micciancio. The hardness of the closest vector problem with preprocessing. *IEEE Transactions on Information Theory*, 47(3):1212–1215, Mar. 2001.

[9] D. Micciancio and S. Goldwasser. *Complexity of lattice problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, Mar. 2002.

[10] O. Regev. Improved Inapproximability of Lattice and Coding Problems with Preprocessing. In *Proceedings of the 18th IEEE annual Conference on Computational Complexity - CCC '03*, pages 315–322, Århus, Denmark, July 2003. IEEE.

[11] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, Mathematische Instituut, Universiry of Amsterdam, 1981. Available on-line at URL http://turing.wins.uva.nl/peter/.