Referee Report: Finding a closest lattice point in a lattice of Voronoi's First Kind

Robby G. McKilliam, Alex Grant, and I. Vaughan L. Clarkson

1 Summary

A set of lattice vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{n+1}$ of an n dimension lattice $\Lambda \subseteq \mathbb{R}^n$, form an *obtuse superbasis* of Λ if $\mathbf{b}_1, \ldots, \mathbf{b}_{n+1}$ generate Λ (i.e. Λ corresponds to all integer combinations of the super basis), $\sum_{i=1}^{n} \mathbf{b}_i = \mathbf{0}$ and $\langle \mathbf{b}_i, \mathbf{b}_j \rangle \leq 0$ for all $i \neq j$. A lattice of Voronoi's first kind is a lattice admitting a *obtuse superbasis*.

This paper gives a novel polynomial time algorithm for solving the Closest Vector Problem (CVP) on lattices of Voronoi's first kind, given an obtuse superbasis of the associated lattice. More precisely, given a target vector $\mathbf{t} \in \mathbb{R}^n$ and a superobtuse basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_{n+1})$ of Λ, they give a polynomial time algorithm for computing a vector $\mathbf{y} \in \Lambda$ such that $\|\mathbf{t} - \mathbf{y}\|_2$ is minimized. This extends a previous algorithm of the first two named authors [7] for solving the Shortest Vector Problem (SVP) on such lattices (that is find a shortest non-zero vector in the lattice). While these lattices are by no means general, Conway and Sloane [2] showed that all lattices of dimension at most 4 are of Voronoi's first kind. From the perspective of quadratic forms, the quadratic forms induced by obtuse superbases are in one to one correspondance with graph Laplacians (this is not explicitly stated in the paper for some reason). Indeed, the quadratic form $\|B\mathbf{x}\|_2^2$ is exactly $\mathbf{x}^\mathsf{T} L_G \mathbf{x}$, where L_G is the Laplacian the weighted graph G on vertices $\{1, \ldots, n+1\}$, with edges weights $w_{ij} = -\langle \mathbf{b}_i, \mathbf{b}_j \rangle \geq 0$, $i \neq j$ (which they call the Selling parameters). More precisely:

$$\|B\mathbf{x}\|_2^2 = \mathbf{x}^\mathsf{T} L_G \mathbf{x} = \sum_{\{i,j\} \in E[G]} w_{ij} (\mathbf{x}_i - \mathbf{x}_j)^2. \tag{1.1}$$

Stated differently, this paper shows that problem of inhomogenous quadratic integer minimization over graph Laplacians, i.e. computing $\arg\min_{\mathbf{x} \in \mathbb{Z}^n} (\mathbf{t} + \mathbf{x})^\mathsf{T} L_G (\mathbf{t} + \mathbf{x})$ for any vector $\mathbf{t}$ and graph G, can be solved in polynomial time. Given the recent frenzy of results in Computer Science relating to Laplacians (fast linear solvers, solving combinatorial optimization problems using electrical flows, etc.), I would not be completely surprised if one couldn't find a nifty combinatorial application of this result.

To present the result from the perspective of Computational Complexity (also not given in the paper), this paper can be viewed as giving a polynomial time algorithm for the Closest Vector Problem with Preprocessing (CVPP) over a special class of lattices. In the preprocessing model, we are allowed to compute any polynomial amount of advice about the lattice (i.e. a good basis, or many short vectors in the lattice, etc.) before answering any CVP queries, where the resources needed to compute this advice are not counted in the runtime of the algorithm. In its *gap decisional version*, where we need only decide whether a target is at distance $\leq d$ or $> \alpha d$ (for some approximation factor $\alpha \geq 1$), it is was shown [1, 4] to be NP-Hard for any constant α, and to not be polynomial time solvable for $\alpha = 2^{\log^{1-\epsilon} n}$, for any fixed $\epsilon > 0$, under the assumption that $NP \subsetneq \mathrm{RTIME}(2^{\log^{O(1/\epsilon)} n})$ (quasi-polynomial time). For its *approximate search version* (i.e. find a lattice vector whose distance to the target is some bounded factor away from optimal), an $O(n^{1.5})$ approximation algorithm was (implicitly) given in [5], which was only recently improved to $O(n/\sqrt{\log n})$ in [3].

Hence, from the standpoint of CVPP, the algorithm in this paper gives essentially the first exact polynomial time algorithm over a reasonably large and natural class of lattices, i.e. those of Voronoi's first kind. Lastly, removing the need for preprocessing for these lattices, at least for solving SVP without being given an obtuse superbasis, would require a substantial breakthrough in the theory of lattice problems (not stated in paper). Indeed, by a simple reduction this would allow us to decide whether a lattice (given any of its bases) is a rotation of $\mathbb{Z}^n$ (which is clearly of Voronoi's first kind). This is a long standing open problem in the theory of lattices, which has only recently been solved in certain very special cases [6].

2 Algorithm and Techniques

At a high level, the presented algorithm is a variant of the *iterative slicer* introduced in [9] specialized for lattices of Voronoi's first kind (given the superobtuse basis as input). The iterative slicer is a method by which one moves closer and closer to a target vector by moving along Voronoi relevant vectors of the lattice, where the Voronoi relevant vectors are the lattice vectors inducing facets of the Voronoi cell (the centrally symmetric polytope corresponding to all points closer to the origin than any other lattice point). Importantly, when one can no longer make progress towards a target vector along Voronoi relevant directions, then the current lattice vector is a closest lattice vector. In [8], a highly sophisticated version of this method was used to give the first $\tilde{O}(2^{2n})$ time and $\tilde{O}(2^n)$ space algorithm for CVP (all previous exact methods ran in $n^{O(n)}$ time).

In the context of this paper, a first crucial observation, due to Conway and Sloane [2], is that given a superobtuse basis $\mathbf{b}_1, \dots, \mathbf{b}_{n+1}$ of Λ, the Voronoi relevant vectors can all be expressed as $0/1$ combinations of the obtuse superbasis. This is a rather special property of these lattices, and we note that it is unknown whether general lattices can have their Voronoi cells expressed in such a "compressed" form (even with very general notions of compressions). Indeed, progress on this front (i.e. showing the existence of good compressed representations), would be a crucial first step to reducing the exponential memory usage of the algorithm in [8].

For their algorithm, given the target $\mathbf{t}$, they proceed as follows:

1. Express $\mathbf{t} = B\mathbf{z}$, $\mathbf{z} \in \mathbb{R}^{n+1}$, where $B = (\mathbf{b}_1, \dots, \mathbf{b}_{n+1})$, and initialize $\mathbf{y} = \lfloor \mathbf{z} \rceil$ as our first guess for the coordinates of the closest vector to $\mathbf{t}$ (under B).

2. Repeat moving from $\mathbf{y}$ to $\mathbf{y} \leftarrow \mathbf{y} + \mathbf{v}$ where

$$\mathbf{v} = \arg\min_{\mathbf{x} \in \{0,1\}^{n+1}} \|B(\mathbf{z} - (\mathbf{y} + \mathbf{x}))\|_2,$$

   as long as we keep getting closer to $\mathbf{t} = B\mathbf{z}$.

3. Return $B\mathbf{y}$.

Here we can see that the algorithm greedily attempts to shrink the distance to the target by moving along the Voronoi relevant direction which makes the most "progress". The main technical part of the paper (and most interesting) is showing that this greedy procedure terminates after at most n steps. Interestingly, it is unknown whether such a procedure converges quickly in general lattices (it probably does not), and we note that the procedure given in [8] converges in $2^n \log n$ iterations.

The second technical part is a small extension of [7], which shows that the computations at each step (i.e. finding the largest progress direction) reduces to computing the min s-t cut of a weighted undirected graph. The graph is an augmentation of the weighted graph G whose Laplacian L_G corresponds to the quadratic form induced by the obtuse superbasis. For more details, we note that by Equation (1.1) the minimization problem $\min_{\mathbf{x} \in \{0,1\}^n \setminus \{\mathbf{0}^n, \mathbf{1}^n\}} \mathbf{x}^T L_G \mathbf{x}$, exactly computes the min cut of the graph G (the main observation of [7] is that this also corresponds to the shortest non-zero vector of Λ). For the one step computations in the above algorithm, we need to solve the inhomogeneous version of the problem, i.e. $\min_{\mathbf{x} \in \{0,1\}^n} (\mathbf{z} + \mathbf{x})^T L_G (\mathbf{z} + \mathbf{x})$ for some $\mathbf{z}$. Here the shift $\mathbf{z}$ adds a linear (i.e. non-quadratic) term to the objective, which the authors show can be accounted for by adding a source and sink to G with well chosen edge weights to the rest of the graph.

Using any standard $O(n^3)$ algorithm for min st-cut, the above gives a full running time of $O(n^4)$ for CVP on these lattices.

3 Recommendation

This paper makes a solid contribution to the theory of lattice problems. I think the results here are well-motivated (I encourage the authors to include some of the motivation mentioned in the summary), and point to the possibility that there might be large and useful classes of lattices where solving CVP is "easy". As mentioned by the authors in the conclusion, it would be very exciting if one could show the existence / give a construction for dense packing lattices for which CVP is polynomially solvable (this would be analoguous to the many constructions of good correcting codes with polynomial decoding algorithms). There is also a non-trivial possibility that the algorithm presented here could have applications in combinatorial optimization.

On a technical level, I found the main convergence proof to be interesting, as I've rarely seen combinatorial arguments successfully applied to lattice problems. On the other hand, while I've convinced myself the authors have the "right" convergence proof, I found its presentation to be rather awkward and unintuitive. I highly recommend that the authors rephrase the proof using the language of Laplacians, which would make things more clear (see the additional comments below for some technical suggestions).

Overall, I am happy to recommend this paper for acceptance.

4 Comments

1. I believe Proposition 2.1 and 2.2 appear essentially verbatim in [8]. Same for Propositions 3.1 (which is essentially by definition) and 3.2, they appear in [9]. I don't think it adds any readibility / anything of value to reprove them here.

2. Almost all the inequalities in the paper can be derived from the following simple observation. For any n vertex weighted graph G with non-negative edge weights, and vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, if there exists a permutation π of $[n]$ such that $\mathbf{x}_{\pi[1]} \leq \cdots \leq \mathbf{x}_{\pi[n]}$ and $\mathbf{y}_{\pi[1]} \leq \cdots \leq \mathbf{y}_{\pi[n]}$ then

   $$\mathbf{x}^{\mathsf{T}} L_G \mathbf{y} = \sum_{\{i,j\} \in E[G]} \mathbf{w}_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{y}_i - \mathbf{y}_j) \geq 0 \,.$$

   Furthermore, if $\mathbf{x}_{\pi[1]} \leq \cdots \leq \mathbf{x}_{\pi[n]}$ and $\mathbf{y}_{\pi[1]} \geq \cdots \geq \mathbf{y}_{\pi[n]}$ then the above inequality is reversed, i.e. $\mathbf{x}^{\mathsf{T}} L_G \mathbf{y} \leq 0$.

   I believe using the above will make many of your proofs more transparent. For example, if you redefine $\mathrm{subtr}(\mathbf{x})$ to return the largest set S such that $\min_{i \in S} \mathbf{x}_i - \max_{j \notin S} \mathbf{x}_j \geq 1$, then notice that if $S \neq \emptyset$ then there is a permutation that puts both $\mathbf{x} - 1_S$ and 1_S in non-decreasing order. In particular this implies that

   $$\mathbf{x}^{\mathsf{T}} L_G \mathbf{x} = (\mathbf{x} - 1_S)^{\mathsf{T}} L_G (\mathbf{x} - 1_S) + 2(\mathbf{x} - 1_S)^{\mathsf{T}} L_G 1_S + 1_S^{\mathsf{T}} L_G 1_S \geq (\mathbf{x} - 1_S)^{\mathsf{T}} L_G (\mathbf{x} - 1_S),$$

   i.e. $\mathbf{x}$ can be made "shorter" by squeezing its components closer together.

   Furthermore, one can use this to give a two line proof of Theorem 4.1. This theorem I actually do think makes sense to reprove here, because modifications of this proof are what drives the convergence bound.

3. Lemma 5.1: there is a typo here. The last identity should read

   $$3.\|B\mathbf{p}\|^2 - \|B(\mathbf{p} + 1_S - 1_T)\|^2 = \Phi(S, p) + \Phi(\bar{T}, \mathbf{p}) + 2\sum_{i \in S} \sum_{j \in S} q_{ij}.$$

   Notice that factor 2 in front of the last term. There are also two related typos on the top of page 12, where sum $\sum_{i \in S} \sum_{j \in S} q_{ij}$ (this appears twice on different lines) should be $2\sum_{i \in S} \sum_{j \in T} q_{ij}$.

4. Lemma 5.7: h is ill-defined if $\mathrm{rng}(\mathbf{v}) = 0$. You should explicitly state that $\mathrm{rng}(\mathbf{v}) \geq 1$.

References

[1] Mikhail Alekhnovich, Subhash Khot, Guy Kindler, and Nisheeth K. Vishnoi. Hardness of approximating the closest vector problem with pre-processing. *Computational Complexity*, 20(4):741–753, 2011.

[2] J. H. Conway and N. J. A. Sloane. Low-dimensional lattices. vi. voronoi reduction of three-dimensional lattices. In *Proceedings: Mathematical and Physical Sciences*, pages 55–68, 1992.

[3] D. Dadush, N. Stephens-Davidowitz, and O. Regev. On the closest vector problem with a distance guarantee. In *Proceedings of the Conference on Computation Complexity*, 2014.

[4] Subhash Khot, Preyas Popat, and Nisheeth K. Vishnoi. $2^{\log^{1-\epsilon} n}$ hardness for closest vector problem with preprocessing. *SIAM Journal on Computing*, 43(3):1184–1205, 2014.

[5] J. C. Lagarias, H. W. Lenstra, Jr., and C.-P. Schnorr. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10(4):333–348, 1990.

[6] H. Lenstra and A. Silverberg. Revisiting the gentry-szydlo algorithm. In *Crypto*, 2014.

[7] R. G. McKilliam and A. Grant. Finding short vectors in a lattice of voronoi's first kind. In *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 2157–2160, 2012.

[8] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. *SIAM Journal on Computing*, 42(3):1364–1391, 2013. Preliminary version in STOC'10.

[9] Naftali Sommer, Meir Feder, and Ofir Shalvi. Finding the closest lattice point by iterative slicing. *SIAM Journal on Discrete Mathematics*, 23(2):715–731, 2009.