1 Further Reading

1.1 Maximum Feasible Linear System (MAXFLS)

Problem 1.1 *Maximum Feasible Linear System (*MAXFLS*)*
INSTANCE: A system of linear equations with coefficient in Z*:* $A \cdot \vec{x} = \vec{b}$.
QUESTION: A maximum subset of the equations that has a solution over Q.

MAXFLS *look* easy since, given matrix A and vector $\vec{b}$ one can, in polynomial time, do the following (by Gaussian elimination):

- Determine if there is a solution, and if so then find one, and if not then produce a certificate of infeasibility.
- If there is no solution then find a $\vec{x}$ such that $A\vec{x}$ is close to $\vec{b}$. More precisely $\vec{x}$ is such that, $A\vec{x} - \vec{b}$ has the *least mean squared error.*

Nevertheless, Amaldi and Kann [3] showed the following:

- The natural decision formulation of MAXFLS is NP-hard.
- Many variants and restrictions of MAXFLS are NP-hard.
- Assume P $\neq$ NP. Many variants of MAXFLS are hard to approximate. The hardness varies with the variant. Some are in APX but not PTAS, and some are harder to approximate than that.

1.2 MAXCUT

Recall the MAXCUT problem has a straightforward $\frac{1}{2}$-approximation algorithm. Better approximation algorithms are known, and lower bounds on approximation are known:

Theorem 1.2

1. *(Goemans & Williamson [6]) There is a* $0.878\ldots$*-approximation algorithm for* MAXCUT *(the number is actually* $\frac{2}{\pi}\min_{0\le\theta\le\pi}\frac{\theta}{1-\cos(\theta)}$*).*
2. *(Hastad [7] building on work of Trevisan et al [14]). Assume* P $\neq$ NP*. Let* $\epsilon > 0$*. there is no* $\frac{16}{17} + \epsilon$*-approximation algorithm for* MAXCUT*. Note that* $\frac{16}{17} \sim 0.941$*.*

Note that if our hardness assumption is P $\neq$ NP then we do not get matching upper and lower bounds. In Chapter **??** we will see that, assuming the Unique Game Conjecture, the algorithm of Goemans & Williamson can be shown to be optimal.

1.3 Closest Vector Problem (CVP)

Def 1.3

1. A lattice $\mathcal{L}$ in R^n is a discrete subgroup of R^n.
2. Let $p \in [1, \infty)$. The *p-norm* of a vector $\vec{x} = (x_1, \ldots, x_n) \in \mathsf{R}^n$ is

$$||\vec{x}||_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p}.$$

   Note that $p = 2$ yields the standard Euclidean distance.

3. If $p = \infty$ then

$$||\vec{x}||_p = \max_{1 \le i \le n} |x_i|.$$

4. The distance between $\vec{x}$ and $\vec{y}$ in norm p is $||\vec{x} - \vec{y}||_p$.

In the next problem let $p \in [1, \infty]$.

Problem 1.4 *Shortest Vector Problem in norm p* (SVP_p)
INSTANCE: A lattice L specified by a basis.
QUESTION: Output the shortest vector in that basis using the p-norm.

Lenstra et al. [11] showed that there is a $2^{n/2}$-approximation to SVP_2. They used it to obtain an algorithm to factor polynomials. Schnorr [13] improved this to a $2^{n(\log\log n)^2/\log n}$-approximation. There are many non-approx results which indicate that the results of the type Lenstra and Schnorr obtained are the best possible. We state some of them and an also refer the reader to the papers cited for earlier results on this topic.

Theorem 1.5

1. *(Boas [15])* SVP_∞ *is* NP*-hard.*
2. *(Ajtai [2])* SVP_2 *is* NP*-hard under randomized reductions.*

3. *(Khot [9]) Assume* NP $\not\subseteq$ RP. *Let* $p \in (1, \infty)$. *There is no poly-time constant-approx for* SVP_p.

4. *(Khot [9]) Assume* NP $\not\subseteq$ RTIME($2^{\mathrm{polylog}(n)}$). *Let* $p \in (1, \infty)$. *Let* $\epsilon > 0$. *There is no poly-time* $2^{(\log n)^{1/2-\epsilon}}$*-approx for* SVP_p.

5. *(Aggarwal et al. [1]) Let* $p \in [1, \infty) - 2\mathsf{Z}$. *Assume* SETH. SVP_p *cannot be solved in time* $O(2^{(1-\epsilon)n})$ *for any* $\epsilon > 0$.

6. *(Haviv & Regev [8] Assume* NP $\not\subseteq$ RTIME($2^{\mathrm{polylog}(n)}$). *Let* $p \in [1, \infty)$. *Let* $\epsilon > 0$. *There is no poly-time* $2^{(\log n)^{1-\epsilon}}$*-approx for* SVP_p.

7. *(Bennett & Peikert [4]) Let* $p \in [1, \infty)$. SVP_p *is* NP*-hard under randomized reductions. This proof has some aspects to it that make derandomizing it plausible. If this is shown then the hardness assumption of* NP $\not\subseteq$ RP *can be changed to* P $\neq$ NP.

Micciancio [12] presented new proofs of the results of Khot [9] and Haviv & Regev [8] that, while still using random reductions, seem likely to be able to derandomize. This gives evidence that (1) Khot's result can be improved to use the hardness assumption P $\neq$ NP, and (2) Haviv & Regev's result can be improved to use the hardness assumption NP $\not\subseteq$ DTIME($2^{\mathrm{polylog}(n)}$).

For most of the results in Theorem 1.5 there are similar, but not identical, upper and lower bounds for CVP, the closest vector problem.

1.4 Minimum Bisection

Problem 1.6 *Minimum Bisection*

INSTANCE: A graph $G = (V, E)$ *on an even number of vertices.*

QUESTION: A partition $V = V_1 \cup V_2$ *such that the number of edges from* V_1 *to* V_2 *is minimized.*

Theorem 1.7

1. *(Feige & Krautghamer [5]) There is a* $O(\log n^2)$*-approximation for Minimum Bisection.*

2. *(Khot [10]) Let* $\epsilon > 0$. *There exists* $\delta > 0$ *(which depends on* ϵ *such that the following is true: Assume* 3-SAT *cannot be solved in* $2^{O(n^\epsilon)}$ *time. Then there is no* $(1 + \delta)$*-approximation for Minimum Bisection.*

References

[1] D. Aggarwal, H. Bennett, A. Golovnev, and N. Stephens-Davidowitz. Fine-grained hardness of CVP(P) - everything that we can prove (and nothing else). In D. Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1816–1835. SIAM, 2021. `https://doi.org/10.1137/1.9781611976465.109`.

[2] M. Ajtai. The shortest vector problem in L_2 is *NP*-hard for randomized reductions (extended abstract). In J. S. Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 10–19. ACM, 1998. `https://doi.org/10.1145/276698.276705`.

[3] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theor. Comput. Sci.*, 147(1&2):181–210, 1995. `https://doi.org/10.1016/0304-3975(94)00254-G`.

[4] H. Bennett and C. Peikert. Hardness of the (approximate) shortest vector problem: A simple proof via Reed-Solomon codes, 2022. `https://arxiv.org/abs/2202.07736`.

[5] U. Feige and R. Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM J. Comput.*, 31(4):1090–1118, 2002. `https://doi.org/10.1137/S0097539701387660`.

[6] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. `https://dl.acm.org/doi/pdf/10.1145/227683.227684`.

[7] J. Hastad. Some optimal inapproximability results. *Journal of the Association of Computing Machinery (JACM)*, 48(4):798–859, 2001. `https://dl.acm.org/doi/10.1145/502090.502098`.

[8] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. *Theory of Computing*, 8(1):513–531, 2012. `https://doi.org/10.4086/toc.2012.v008a023`.

[9] S. Khot. Hardness of approximating the shortest vector problem in lattices. *J. ACM*, 52(5):789–808, 2005.
`https://doi.org/10.1145/1089023.1089027`.

[10] S. Khot. Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.
`https://epubs.siam.org/doi/10.1137/S0097539705447037`.

[11] A. Lenstra, H. Lenstra, and L. Lovasz. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:513–534, 1982.

[12] D. Micciancio. Inapproximability of the shortest vector problem: Toward a deterministic reduction. *Theory of Computing*, 8(1):487–512, 2012.
`https://doi.org/10.4086/toc.2012.v008a022`.

[13] C. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.
`https://doi.org/10.1016/0304-3975(87)90064-8`.

[14] L. Trevisan, G. B. Sorkin, M. Sudan, and D. P. Williamson. Gadgets, approximation, and linear programming. *SIAM J. Comput.*, 29(6):2074–2097, 2000.
`https://doi.org/10.1137/S0097539797328847`.

[15] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Tech Report 8104, University of Amsterdam, Dept of Mathematics, Neverthlands, 1981.