Groverize Monotone Local Search. (Short Note)

David Ponarovsky

April 25, 2023

Abstract

In this paper, we improve a wide range of upper bounds for a variety of **NP** problems, by plugging Grover into the work made by [Fom+15]. We emphasize that this work has only a technical value and does not present any new idea. Nevertheless, we think it is worth sharing how easy and straightforwardly integrating quantum might be. We coin the term Groverize, harvesting a quantum improvement by doing almost nothing.

1 Todo.

1. Write the table (sage script).
2. Add definitions. Problem description.
3. Complete the 'proof'.
4. Prove lower bound.

2 Introduction.

We follow the study of [Fom+15], who relate between the parametrized complexity to the general average case complexity. Crudely put, they shown that for particular wide range of **NP** hard problems, a solution which run exponentially at some complexity parameter, for example the tree-width of a graph, can be used to derive a batter than bruteforce solution for the general problem. We continue their work by plugin the Grover search [Gro96] routine instead the original sampling process. We will simplify the definitions given at [Fom+15] and use the following definitions instead:

A decision problem is said to have a parameterized algorithm if there is a mapping between its instances and the natural number k such that there exists an algorithm that solves the problem in running time that is exponential in k and polynomial in n.

We will say that a problem having a parametrized algorithms is an *extension problem* if for any instance of the problem P, requiring any of the input bits to be 1 can be reduced to another instance of the problem P' such that $\phi\left(P'\right) = \phi\left(P\right) - 1$. For example, consider **3-SAT** with the restriction that the Hamming weight of the assignment would be at most k. Fixing an arbitrary bit x_i to be 1 can be reduced to another 3-**SAT** formula by erasing any of the clauses containing x_i and replacing any of the occurrences of $\bar{x}_i$ by another terminal on the same clause (i.e. $\bar{x}_i \wedge \bar{y} \wedge z \mapsto \bar{y} \wedge \bar{y} \wedge z$). Now, note that an assignment that satisfies the new formula at Hamming weight at most $k-1$ combined with $x_i \leftarrow 1$ is an assignment to the original formula at weight at most k. Given the fact that we have a brute-force algorithm which tries all the partitions in time roughly $\mathcal{O}\left(n^k\right)$, it follows that this problem is an extension problem. Let us state the above:

Definition 1 (Parameterized Computable)**.** *Let L be a problem family, and use the notation $P \in L$ to indicate that P is an instance of L (e.g., if L is a 3-SAT problem), and by $|P|$ to denote the length of the binary string encoding P. L is said to be parameterized computable if there exists a mapping $\phi : L \rightarrow \mathbb{N}$ and an algorithm $\mathcal{A}$ such that:*

1. $\phi(P) \in [|P|]$ *for any* $P \in L$
2. $\mathcal{A}$ *returns on* P *at running time* $\mathcal{O}\left(c^{\phi(P)} \cdot Poly(|P|)\right)$ *for a fixed* c *which does not depend on* P.

Definition 2 (Extendable Problem)**.** *A problem family* L *which is parametrized computable will be said an Extendable problem, if for any* $P \in L$*, the problem obtained from* P *by setting one of the input bits to be* 1 *can be converted by polynomial-time reduction to another instance of* L*, denoted as* $P' \in L$*, such that* $\phi(P') = \phi(P) - 1$.

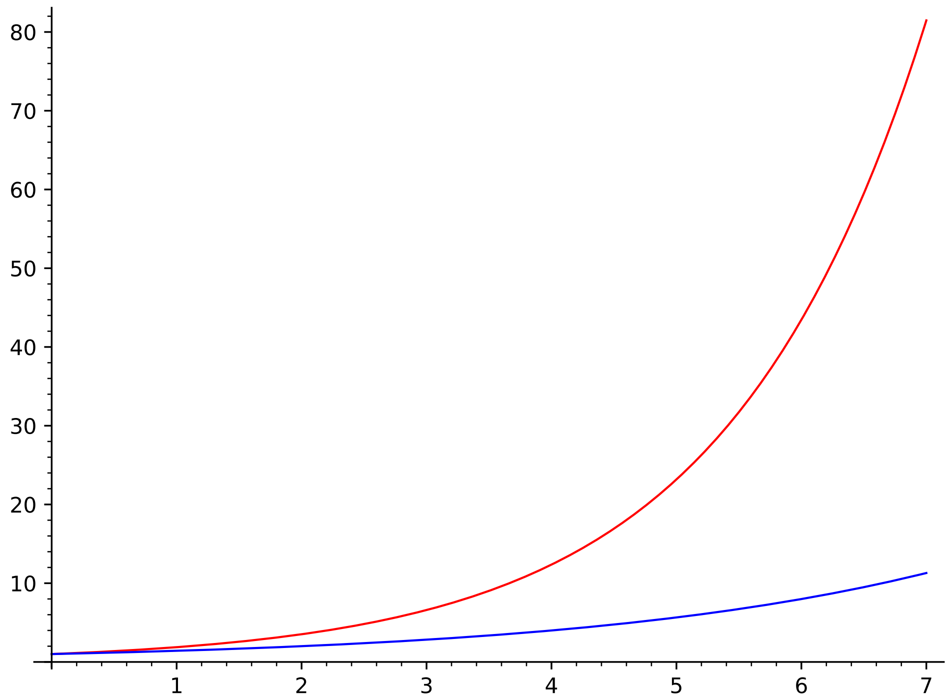
Theorem 1 (Classic Monotone Search.)**.** *Let* L *be an Extendable Problem with constant* c*. Then there is an randomized algorithm that compute* L *at cost* $\left(2 - \frac{1}{c}\right)^n N^{\mathcal{O}(1)}$

Theorem 2 (Quantum Monotone Search.)**.** *Let* L *be an Extendable Problem with constant* c*. Then there is an quantum algorithm that compute* L *at cost* $\left(2 - \frac{1}{c^2}\right)^{\frac{n}{2}} N^{\mathcal{O}(1)}$

3 Local Monotone Search.

One can consider local search as a weighted tiling problem, in a sense that we would like to cover all the space. For instance, in the non-parameterized setting, or in the query model, one has no chance to beat the 'scanning all the options' approach. However, having an oracle gives us a degree of freedom to sample points from a sparser lattice and then perform a 'cheap' computation to scan the ball around it. To gain a better understanding, let us consider the extreme case. If it is guaranteed that $\mathcal{A}$ returns a solution at radius $\frac{1}{4}n$ in α time, then instead of sampling all the possible inputs, one can sample just from all the points on the hypercube such that any two different points are at least $\frac{3}{4}n$ apart and pay a total of $\sim O\left(2^{\frac{3}{4}n}\alpha\right)$ time to cover all the options. When the problem has a monotone property definition 2, the idea remains the same, but now we aim to fill the space with cones instead of balls. It turns out that computing the optimal ratio of the lattice to sample from is a bit tricky; this is exactly what [Fom+15] did in their work.

$$\sum_{k' \leq k} \frac{1}{\sqrt{p(k')}} \cdot c^{k'-t} N^{\mathcal{O}(1)} \leq \max_{k' \leq k} \left(\frac{\binom{n-|X|}{t}}{\binom{k'}{t}} \right)^{\frac{1}{2}} \cdot c^{k'-t} N^{\mathcal{O}(1)} =$$
$$\left(\max_{k' \leq k} \frac{\binom{n-|X|}{t}}{\binom{k'}{t}} \cdot c^{2\left(k'-t\right)} \right)^{\frac{1}{2}} N^{\mathcal{O}(1)} = \left(\max_{k \leq n-|X|} \frac{\binom{n-|X|}{t}}{\binom{k}{t}} \right)^{\frac{1}{2}} \cdot c^{2(k-t)} N^{\mathcal{O}(1)} \leq$$
$$\Rightarrow \left(2 - \frac{1}{c^2}\right)^{\frac{n-|X|}{2}} N^{\mathcal{O}(1)}$$



Problem Name	Parameterized	Groverize	New bound	Previous Bound
Feedback Vertex Set	3^k (r) [Cyg+11]	1.3744^k	1.6667^n (r)	
Feedback Vertex Set	3.592^k [KP14]	1.3865^k	1.7217^n	1.7347^n [FTV1
Subset Feedback Vertex Set	4^k [**Wahlstrom14**]	1.3919^k	1.7500^n	1.8638^n [Fom+1
Feedback Vertex Set in Tournaments	1.6181^k [KL16]	1.2720^k	1.3820^n	1.4656^n [KL1
Group Feedback Vertex Set	4^k [**Wahlstrom14**]	1.3919^k	1.7500^n	NPR
Node Unique Label Cover	$\lvert\Sigma\rvert^{2k}$ [**Wahlstrom14**]	1.3919^k	$(2-\frac{1}{\lvert\Sigma\rvert^2})^n$	NPR
Vertex (r,ℓ)-Partization $(r,\ell \leq 2)$	3.3146^k [**KolayP15**; Bas+17]	1.3817^k	1.6984^n	NPR
Interval Vertex Deletion	8^k [Cao16]	1.3466^k	1.8750^n	$(2-\varepsilon)^n$ for $\varepsilon < 10^{-20}$ [BFP1
Proper Interval Vertex Deletion	6^k [tV13; Cao15]	1.4087^k	1.8334^n	$(2-\varepsilon)^n$ for $\varepsilon < 10^{-20}$ [BFP1
Block Graph Vertex Deletion	4^k [Agr+16]	1.4044^k	1.7500^n	$(2-\varepsilon)^n$ for $\varepsilon < 10^{-20}$ [BFP1
Cluster Vertex Deletion	1.9102^k [Bor+14]	1.3919^k	1.4765^n	1.6181^n [Fom+1
Thread Graph Vertex Deletion	8^k [Kan+15]	1.3919^k	1.8750^n	NPR
Multicut on Trees	1.5538^k [Kan+14]	1.3138^k	1.3565^n	NPR
3-Hitting Set	2.0755^k [**MagnusPhD07**]	1.4087^k	1.5182^n	1.6278^n [**MagnusPhD0**
4-Hitting Set	3.0755^k [Fom+10]	1.2593^k	1.6750^n	1.8704^n [Fom+1
d-Hitting Set $(d \geq 3)$	$(d-0.9245)^k$ [Fom+10]	1.1763^k	$(2-\frac{1}{(d-0.9245)})^n$	[Coc+16; Fom+1
Min-Ones 3-SAT	2.562^k [**abs-1007-1166**]	1.3296^k	1.6097^n	NPR
Min-Ones d-SAT $(d \geq 4)$	d^k	1.3763^k	$(2-\frac{1}{d})^n$	NPR
Weighted d-SAT $(d \geq 3)$	d^k	1.3763^k	$(2-\frac{1}{d})^n$	NPR
Weighted Feedback Vertex Set	3.6181^k [Agr+16]	1.1763^k	1.7237^n	1.8638^n [Fom+0
Weighted 3-Hitting Set	2.168^k [SZ15]	1.3593^k	1.5388^n	1.6755^n [Coc+1
Weighted d-Hitting Set $(d \geq 4)$	$(d-0.832)^k$ [Fom+10; SZ15]	1.3919^k	$(2-\frac{1}{d-0.932})^n$	[Coc+1

Table 1: Summary of known and new results for different optimization problems. NPR means that we are not aware of any previous algorithms faster than brute-force. All bounds suppress factors polynomial in the input size N. The algorithms in the first row are randomized (r).

References

[Gro96] Lov K. Grover. *A fast quantum mechanical algorithm for database search*. 1996. arXiv: quant-ph/9605043 [quant-ph].

[Fom+08] Fedor V. Fomin et al. “On the minimum feedback vertex set problem: exact and enumeration algorithms”. In: *Algorithmica* 52.2 (2008), pp. 293–307. ISSN: 0178-4617. DOI: 10.1007/s00453-007-9152-0. URL: https://doi.org/10.1007/s00453-007-9152-0.

[Fom+10] Fedor V. Fomin et al. “Iterative compression and exact algorithms”. In: *Theoret. Comput. Sci.* 411.7-9 (2010), pp. 1045–1053. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2009.11.012. URL: https://doi.org/10.1016/j.tcs.2009.11.012.

[Cyg+11] Marek Cygan et al. “Solving connectivity problems parameterized by treewidth in single exponential time (extended abstract)”. In: *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science—FOCS 2011*. IEEE Computer Soc., Los Alamitos, CA, 2011, pp. 150–159. DOI: 10.1109/FOCS.2011.23. URL: https://doi.org/10.1109/FOCS.2011.23.

[tV13] Pim van ’t Hof and Yngve Villanger. “Proper interval vertex deletion”. In: *Algorithmica* 65.4 (2013), pp. 845–867. ISSN: 0178-4617. DOI: 10.1007/s00453-012-9661-3. URL: https://doi.org/10.1007/s00453-012-9661-3.

[BFP13] Ivan Bliznets, Fedor V. Fomin, and Yngve Pilipczuk Michałand Villanger. “Largest chordal and interval subgraphs faster than 2^n”. In: *Algorithms—ESA 2013*. Vol. 8125. Lecture Notes in Comput. Sci. Springer, Heidelberg, 2013, pp. 193–204. DOI: 10.1007/978-3-642-40450-4_17. URL: https://doi.org/10.1007/978-3-642-40450-4_17.

[Bor+14] Anudhyan Boral et al. “A fast branching algorithm for cluster vertex deletion”. In: *Computer science—theory and applications*. Vol. 8476. Lecture Notes in Comput. Sci. Springer, Cham, 2014, pp. 111–124. DOI: 10.1007/978-3-319-06686-8_9. URL: https://doi.org/10.1007/978-3-319-06686-8_9.

[Fom+14] Fedor V. Fomin et al. “Enumerating minimal subset feedback vertex sets”. In: *Algorithmica* 69.1 (2014), pp. 216–231. ISSN: 0178-4617. DOI: 10.1007/s00453-012-9731-6. URL: https://doi.org/10.1007/s00453-012-9731-6.

[Kan+14] Iyad Kanj et al. “Algorithms for cut problems on trees”. In: *Combinatorial optimization and applications*. Vol. 8881. Lecture Notes in Comput. Sci. Springer, Cham, 2014, pp. 283–298. DOI: 10.1007/978-3-319-12691-3_22. URL: https://doi.org/10.1007/978-3-319-12691-3_22.

[KP14] Tomasz Kociumaka and Marcin Pilipczuk. “Faster deterministic Feedback Vertex Set”. In: *Inform. Process. Lett.* 114.10 (2014), pp. 556–560. ISSN: 0020-0190. DOI: 10.1016/j.ipl.2014.05.001. URL: https://doi.org/10.1016/j.ipl.2014.05.001.

[Cao15] Yixin Cao. “Unit interval editing is fixed-parameter tractable”. In: *Automata, languages, and programming. Part I*. Vol. 9134. Lecture Notes in Comput. Sci. Springer, Heidelberg, 2015, pp. 306–317. DOI: 10.1007/978-3-662-47672-7_25. URL: https://doi.org/10.1007/978-3-662-47672-7_25.

[FTV15] Fedor V. Fomin, Ioan Todinca, and Yngve Villanger. “Large induced subgraphs via triangulations and CMSO”. In: *SIAM J. Comput.* 44.1 (2015), pp. 54–87. ISSN: 0097-5397. DOI: 10.1137/140964801. URL: https://doi.org/10.1137/140964801.

[Fom+15] Fedor V. Fomin et al. *Exact Algorithms via Monotone Local Search*. 2015. arXiv: 1512.01621 [cs.DS].

[Kan+15] Mamadou Moustapha Kanté et al. “An FPT algorithm and a polynomial kernel for linear rankwidth-1 vertex deletion”. In: *10th International Symposium on Parameterized and Exact Computation*. Vol. 43. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2015, pp. 138–150.

[SZ15] Hadas Shachnai and Meirav Zehavi. “A multivariate approach for weighted FPT algorithms”. In: *Algorithms—ESA 2015*. Vol. 9294. Lecture Notes in Comput. Sci. Springer, Heidelberg, 2015, pp. 965–976. DOI: 10.1007/978-3-662-48350-3_80. URL: https://doi.org/10.1007/978-3-662-48350-3_80.

[Agr+16] Akanksha Agrawal et al. “A faster FPT algorithm and a smaller kernel for block graph vertex deletion”. In: *LATIN 2016: theoretical informatics*. Vol. 9644. Lecture Notes in Comput. Sci. Springer, Berlin, 2016, pp. 1–13. DOI: 10.1007/978-3-662-49529-2_1. URL: https://doi.org/10.1007/978-3-662-49529-2_1.

[Cao16] Yixin Cao. “Linear recognition of almost interval graphs”. In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 2016, pp. 1096–1115. DOI: 10.1137/1.9781611974331.ch77. URL: https://doi.org/10.1137/1.9781611974331.ch77.

[Coc+16] Manfred Cochefert et al. “Faster algorithms to enumerate hypergraph transversals”. In: *LATIN 2016: theoretical informatics*. Vol. 9644. Lecture Notes in Comput. Sci. Springer, Berlin, 2016, pp. 306–318. DOI: 10.1007/978-3-662-49529-2_23. URL: https://doi.org/10.1007/978-3-662-49529-2_23.

[KL16] Mithilesh Kumar and Daniel Lokshtanov. “Faster exact and parameterized algorithm for feedback vertex set in tournaments”. In: *33rd Symposium on Theoretical Aspects of Computer Science*. Vol. 47. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016, Art. No. 49, 13.

[Bas+17] Julien Baste et al. “Parameterized complexity dichotomy for (r,ℓ)-vertex deletion”. In: *Theory Comput. Syst.* 61.3 (2017), pp. 777–794. ISSN: 1432-4350. DOI: 10.1007/s00224-016-9716-y. URL: https://doi.org/10.1007/s00224-016-9716-y.