Quantum algorithms (CO 781/CS 867/QIC 823, Winter 2011)
Andrew Childs, University of Waterloo

# LECTURE 10: Kuperberg's algorithm for the dihedral HSP

In this lecture, we will discuss a quantum algorithm for the dihedral hidden subgroup problem. No polynomial-time algorithm for this problem is known. However, Kuperberg gave a quantum algorithm that runs in subexponential (though superpolynomial) time—specifically, it runs in time $2^{O(\sqrt{\log |G|})}$.

## The HSP in the dihedral group

The dihedral group of order $2N$, denoted $D_N$, is the group of symmetries of a regular $N$-gon. It has the presentation

$$D_N = \langle r, s \colon r^2 = s^N = 1,\ rsr = s^{-1} \rangle. \tag{1}$$

Here $r$ can be thought of as a reflection about some fixed axis, and $s$ can be thought of as a rotation of the $N$-gon by an angle $2\pi/N$.

Using the defining relations, we can write any group element in the form $s^x r^a$ where $x \in \mathbb{Z}_N$ and $a \in \mathbb{Z}_2$. Thus we can equivalently think of the group as consisting of elements $(x, a) \in \mathbb{Z}_N \times \mathbb{Z}_2$. Since

$$\begin{aligned} (s^x r^a)(s^y r^b) &= s^x r^a s^y r^a r^{a+b} && (2)\\ &= s^x s^{(-1)^a y} r^{a+b} && (3)\\ &= s^{x+(-1)^a y} r^{a+b}, && (4) \end{aligned}$$

the group operation '$\cdot$' on such elements can be expressed as

$$(x, a) \cdot (y, b) = (x + (-1)^a y, a + b). \tag{5}$$

(In particular, this shows that the dihedral group is the semidirect product $\mathbb{Z}_N \rtimes_\varphi \mathbb{Z}_2$, where $\varphi : \mathbb{Z}_2 \to \mathrm{Aut}(\mathbb{Z}_N)$ is defined by $\varphi(a)(y) = (-1)^a y$.) It is also easy to see that the group inverse is

$$(x, a)^{-1} = (-(-1)^a x, a). \tag{6}$$

The subgroups of $D_N$ are either cyclic or dihedral. The possible cyclic subgroups are of the form $\langle (x, 0) \rangle$ where $x \in \mathbb{Z}_N$ is either 0 or some divisor of $N$. The possible dihedral subgroups are of the form $\langle (y, 1) \rangle$ where $y \in \mathbb{Z}_N$, and of the form $\langle (x, 0), (y, 1) \rangle$ where $x \in \mathbb{Z}_N$ is some divisor of $N$ and $y \in \mathbb{Z}_x$. A result of Ettinger and Høyer reduces the general dihedral HSP, in which the hidden subgroup could be any of these possibilities, to the dihedral HSP with the promise that the hidden subgroup is of the form $\langle (y, 1) \rangle = \{(0, 0), (y, 1)\}$, i.e., a subgroup of order 2 generated by the reflection $(y, 1)$.

The basic idea of the Ettinger-Høyer reduction is as follows. Suppose that $f : D_N \to S$ hides a subgroup $H = \langle (x, 0), (y, 1) \rangle$. Then we can consider the function $f$ restricted to elements from the abelian group $\mathbb{Z}_N \times \{0\} \leq D_N$. This restricted function hides the subgroup $\langle (x, 0) \rangle$, and since the restricted group is abelian, we can find $x$ efficiently using Shor's algorithm. Now $\langle (x, 0) \rangle \trianglelefteq D_N$ (since $(z, a)(x, 0)(z, a)^{-1} = (z + (-1)^a x, a)(-(-1)^a z, a) = ((-1)^a x, 0) \in \mathbb{Z}_N \times \{0\}$), so we can define the quotient group $D_N / \langle (x, 0) \rangle$. But this is simply a dihedral group (of order $N/x$), and if we now define a function $f'$ as $f$ evaluated on some coset representative, it hides the subgroup $\langle (y, 1) \rangle$. Thus, in the rest of this lecture, we will assume that the hidden subgroup is of the form $\langle (y, 1) \rangle$ for some $y \in \mathbb{Z}_N$ without loss of generality.

## Fourier sampling in the dihedral group

When the hidden subgroup is $H = \langle (y, 1) \rangle$, one particular left transversal of $H$ in $G$ consists of the left coset representatives $(z, 0)$ for all $z \in \mathbb{Z}_N$. The coset state corresponding to the coset $(z, 0)H$ is

$$|(z, 0)\{(0, 0), (y, 1)\}\rangle = \frac{1}{\sqrt{2}}(|z, 0\rangle + |y + z, 1\rangle). \tag{7}$$

We would like to determine $y$ using samples of this state.

We have seen that to distinguish coset states in general, one should start by performing weak Fourier sampling: apply a Fourier transform over $G$ and then measure the irrep label. However, in this case we will instead simply Fourier transform the first register over $\mathbb{Z}_N$, leaving the second register alone. It is possible to show that measuring the first register of the resulting state is essentially equivalent to performing weak Fourier sampling over $D_N$ (and discarding the row register), but for simplicity we will just consider the abelian procedure.

Fourier transforming the first register over $\mathbb{Z}_N$, we obtain

$$(F_{\mathbb{Z}_N} \otimes I_2)|(z, 0)H\rangle = \frac{1}{\sqrt{2N}} \sum_{k \in \mathbb{Z}_N} (\omega_N^{kz}|k, 0\rangle + \omega_N^{k(y+z)}|k, 1\rangle \tag{8}$$

$$= \frac{1}{\sqrt{N}} \sum_{k \in \mathbb{Z}_N} \omega_N^{kz}|k\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + \omega_N^{ky}|1\rangle). \tag{9}$$

If we then measure the first register, we obtain one of the $N$ values of $k$ uniformly at random, and we are left with the post-measurement state

$$|\psi_k\rangle := \frac{1}{\sqrt{2}}(|0\rangle + \omega_N^{yk}|1\rangle). \tag{10}$$

Thus we are left with the problem of determining $y$ given the ability to produce single-qubit states $|\psi_k\rangle$ of this form (where $k$ is known).

## Combining states

It would be very useful if we could prepare states $|\psi_k\rangle$ with particular values of $k$. For example, if we could prepare the state $|\psi_{N/2}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^y|1\rangle)$, then we could learn the parity of $y$ (i.e., its least significant bit) by measuring in the basis of states $|\pm\rangle := (|0\rangle \pm |1\rangle)/\sqrt{2}$. The main idea of Kuperberg's algorithm is to combine states of the form (10) to produce new states of the same form, but with more desirable values of $k$.

To combine states, we can use the following procedure. Given two states $|\psi_p\rangle$ and $|\psi_q\rangle$, perform a controlled-not gate from the former to the latter, giving

$$|\psi_p, \psi_q\rangle = \frac{1}{2}(|0, 0\rangle + \omega_N^{yp}|1, 0\rangle + \omega_N^{yq}|0, 1\rangle + \omega_N^{y(p+q)}|1, 1\rangle) \tag{11}$$

$$\mapsto \frac{1}{2}(|0, 0\rangle + \omega_N^{yp}|1, 1\rangle + \omega_N^{yq}|0, 1\rangle + \omega_N^{y(p+q)}|1, 0\rangle) \tag{12}$$

$$= \frac{1}{\sqrt{2}}(|\psi_{p+q}, 0\rangle + \omega_N^{yq}|\psi_{p-q}, 1\rangle). \tag{13}$$

Then a measurement on the second qubit leaves the first qubit in the state $|\psi_{p\pm q}\rangle$ (up to an irrelevant global phase), with the $+$ sign occurring when the outcome is 0 and the $-$ sign occurring when the outcome is 1, each outcome occurring with probability $1/2$.

This combination operation has a nice representation-theoretic interpretation: the state indices $p$ and $q$ can be viewed as labels of irreducible representations of $D_N$, and the extraction of $|\psi_{p\pm q}\rangle$ can be viewed as decomposing their tensor product (a reducible representation of $D_N$) into one of two irreducible components.

### The Kuperberg sieve

Now we are ready to describe how the algorithm works. For simplicity, we will assume from now on that $N = 2^n$ is a power of 2. For such a dihedral group, it is actually sufficient to be able to determine the least significant bit of $y$, since such an algorithm could be used recursively to determine all the bits of $y$. This can be seen as follows. The group $D_N$ contains two subgroups isomorphic to $D_{N/2}$, namely $\{(2x, 0), (2x, 1) : x \in \mathbb{Z}_{N/2}\}$ and $\{(2x, 0), (2x + 1, 1) : x \in \mathbb{Z}_{N/2}\}$. The hidden subgroup is a subgroup of the former if $y$ has even parity, and of the latter if $y$ has odd parity. Thus, once we learn the parity of $y$, we can restrict our attention to the appropriate $D_{N/2}$ subgroup. The elements of either $D_{N/2}$ subgroup can be represented using only $n - 1$ bits, and finding the least significant bit of the hidden reflection within this subgroup corresponds to finding the second least significant bit of $y$ in $D_N$. Continuing in this way, we can learn all the bits of $y$ with only $n$ iterations of an algorithm for finding the least significant bit of the hidden reflection.

The idea of Kuperberg's algorithm is to start with a large number of states, and collect them into pairs $|\psi_p\rangle, |\psi_q\rangle$ that share many of their least significant bits, such that $|\psi_{p-q}\rangle$ is likely to have many of its least significant bits equal to zero. Trying to zero out all but the most significant bit in one shot would require an exponential running time, so instead we will proceed in stages, only trying to zero some of the least significant bits in each stage; this will turn out to give an improvement.

Specifically, the algorithm proceeds as follows:

1. Prepare $\Theta(16^{\sqrt{n}})$ coset states of the form (10), where each copy has $k \in \mathbb{Z}_{2^n}$ chosen independently and uniformly at random.
2. For each $j = 0, 1, \ldots, m - 1$ where $m := \lceil \sqrt{n} \rceil$, assume the current coset states are all of the form $|\psi_k\rangle$ with at least $mj$ of the least significant bits of $k$ equal to 0. Collect them into pairs $|\psi_p\rangle, |\psi_q\rangle$ that share at least $m$ of the next least significant bits, discarding any qubits that cannot be paired. Create a state $|\psi_{p\pm q}\rangle$ from each pair, and discard it if the $+$ sign occurs. Notice that the resulting states have at least $m(j + 1)$ significant bits equal to 0.
3. The remaining states are of the form $|\psi_0\rangle$ and $|\psi_{2^{n-1}}\rangle$. Measure one of the latter states in the $|\pm\rangle$ basis to determine the least significant bit of $y$.

Since this algorithm requires $2^{O(\sqrt{n})}$ initial queries and proceeds through $O(\sqrt{n})$ stages, each of which takes at most $2^{O(\sqrt{n})}$ steps, the overall running time is $2^{O(\sqrt{n})}$.

### Analysis of the Kuperberg sieve

To show that this algorithm works, we need to prove that some qubits survive to the final stage of the process with non-negligible probability. Let's analyze a more general version of the algorithm to see why we should try to zero out $\sqrt{n}$ bits at a time, starting with $2^{O(\sqrt{n})}$ states.

Suppose we try to cancel $m$ bits in each stage, so that there are $n/m$ stages (not yet assuming any relationship between $m$ and $n$), starting with $2^\ell$ states. Each combination operation succeeds with probability $1/2$, and turns 2 states into 1, so at each step we retain only about $1/4$ of the

states that can be paired. Now when we pair states that allow us to cancel $m$ bits, there can be at most $2^m$ unpaired states, since that is the number of values of the $m$ bits to be canceled. Thus if we ensure that there are at least $2 \cdot 2^m$ states at each stage, we expect to retain at least a $1/8$ fraction of the states for the next stage. Since we begin with $2^\ell$ states,we expect to have at least $2^{\ell-3j}$ states left after the $j$th stage. Thus, to have $2 \cdot 2^m$ states remaining at the last stage of the algorithm, we require $2^{\ell-3n/m} > 2^{m+1}$, or $\ell > m + 3n/m + 1$. This is minimized by choosing $m \approx \sqrt{n}$, so we see that $\ell \approx 4\sqrt{n}$ suffices.

This analysis is not quite correct because we do not obtain precisely a $1/8$ fraction of the paired states for use in the next stage. For most of the stages, we have many more than $2 \cdot 2^m$ states, so nearly all of them can be paired, and the expected fraction remaining for the next stage is close to $1/4$. Of course, the precise fraction will experience statistical fluctuations. However, since we are working with a large number of states, the deviations from the expected values are very small, and a more careful analysis (using the Chernoff bound) shows that the procedure succeeds with high probability. For a detailed argument, see section 3.1 of Kuperberg's paper (SICOMP version). That paper also gives an improved algorithm that runs faster and that works for general $N$.

Note that this algorithm uses not only superpolynomial time, but also superpolynomial space, since all $\Theta(16^{\sqrt{n}})$ coset states are present at the start of the algorithm. However, by creating a smaller number of coset states at a time and combining them according to the solution of a subset sum problem, Regev showed how to make the space requirement polynomial with only a slight increase in the running time.

### Entangled measurements

Although this algorithm acts on pairs of coset states at a time, the overall algorithm effectively implements a highly entangled measurement on all $\Theta(\sqrt{16^n})$ registers, since the combination operation that produces $|\psi_{p\pm q}\rangle$ entangles the coset states $|\psi_p\rangle$ and $|\psi_q\rangle$. The same is true of Regev's polynomial-space variant.

It is natural to ask whether a similar sieve could be applied to other hidden subgroup problems, such as in the symmetric group, for which highly entangled measurements are necessary. Alagic, Moore, and Russell used a similar approach to give a subexponential-time algorithm for the hidden subgroup problem in the group $G^n$, where $G$ is a fixed non-Abelian group. (Note that the HSP in $G^n$ can be much harder than solving $n$ instances of the HSP in $G$, since $G^n$ has many subgroups that are not direct products of subgroups of $G$.) But unfortunately, this kind of sieve does not seem well-suited to the symmetric group. In particular, Moore, Russell, and Sniady gave the following negative result for the HSP in $S_n \wr \mathbb{Z}_2$, where the hidden subgroup is promised to be either trivial or an involution. Consider any algorithm that works by combining pairs of hidden subgroup states to produce a new state in the decomposition of their tensor product into irreps (i.e., in their *Clebsch-Gordan decomposition*), and uses the sequence of measurement results to guess whether the hidden subgroup is trivial or nontrivial. Any such algorithm must use $2^{\Omega(\sqrt{n})}$ queries. Thus it is not possible to give a significantly better-than-classical algorithm for graph isomorphism in this way, since there are classical algorithms for graph isomorphism that run in time $2^{O(\sqrt{n/\log n})}$.

Note that entangled measurements are *not* information-theoretically necessary for the dihedral HSP: Ettinger and Høyer gave an explicit measurement (i.e., an explicit basis for strong Fourier sampling) from which the measurement results give sufficient information to determine the hidden subgroup. Suppose that, given the state (10), we simply measure in the $|\pm\rangle$ basis. Then we obtain

the result $|+\rangle$ with probability

$$\left|\left(\frac{\langle 0| + \langle 1|}{\sqrt{2}}\right)\left(\frac{|0\rangle + \omega_N^{yk}|1\rangle}{\sqrt{2}}\right)\right|^2 = \left|\frac{1+\omega_N^{yk}}{2}\right|^2 = \cos^2\frac{\pi yk}{N}. \tag{14}$$

If we postselect on obtaining this outcome (which happens with probability $1/2$ over the uniformly random value of $k$, assuming $y \neq 0$), then we effectively obtain each value $k \in \mathbb{Z}_N$ with probability $\Pr(k|+) = \frac{2}{N}\cos^2\frac{\pi yk}{N}$. It is not hard to show that these distributions are statistically far apart for different values of $k$, so that they can in principle be distinguished with only polynomially many samples. However, no efficient (or even subexponential time) classical (or even quantum) algorithm for doing so is known.