Inside the Black Box: Revisiting the Bernstein-Vazirani Problem

N. David Mermin[1] *

[1] *Department of Physics, Cornell University.*
Ithaca, NY 14853-2501, USA.

Abstract. The black-boxed oracle of the Bernstein-Vazirani problem can be trivially imitated by a set of cNOT gates. From this perspective, the solution is an obvious consequence of the ability of Hadamard gates to exchange the target and control qubits of a cNOT gate. Viewed in this way the solution does not even hint at quantum parallelism. The only role quantum mechanics plays is through its (remarkable) ability to reverse the action of a cNOT gate by means of *local* operations on the target and control qubits.

Keywords: Bernstein-Vazirani problem, quantum parallelism

Let $f_a(x) = x \cdot a$ be a function from n bit integers to $\{0,1\}$, where a is a fixed n bit integer and $x \cdot a$ is the bitwise modulo 2 inner product: $x \cdot a = x_0 a_0 \oplus \cdots \oplus x_{n-1} a_{n-1}$, where $\oplus$ indicates addition modulo 2. We are given a unitary black-boxed oracle U_a that takes the state of an n-qubit input register and one-qubit output register, $|\,x\rangle_n |\,y\rangle_1$, into $|\,x\rangle_n |\,y \oplus f_a(x)\rangle_1$. The Bernstein-Vazirani problem is to determine the integer a with the least number of invocations of the oracle. With a classical computer one can learn the j-th bit of a by applying the oracle to $x = 2^j$, thereby requiring n invocations to find the n-bit number a. Since any other classical input x will give some modulo-2 linear combination of the bits of a, and since one needs n independent such linear combinations to determine a, n invocations is the best you can do classically. But with a quantum computer a single call to the oracle suffices.

The conventional demonstration of this wonderful ability of a quantum computer starts by applying an n-fold Hadamard transformation to the input register (initially set to all 0's) to convert it to the familiar equally weighted superposition of all 2^n possible inputs. ("Massive quantum parallelism.") One then puts 1 into the output register and applies one more Hadamard to it, thereby converting the shift by $f_a(x)$ into multiplication by $(-1)^{f_a(x)}$. Then the oracle U_a is called, and then another n-fold Hadamard is applied to the input register. The combined effect of all this is not hard to work out, and one finds that there is a very convenient complete destructive interference leading to the vanishing of every single term in the final state of the input register except for $|\,a\rangle_n$. So with one invocation of U_a, preceded and followed by the application of Hadamards, starting with the initial state $|\,0\rangle_n |\,1\rangle_1$, one ends up with the input register in the state $|\,a\rangle_n$ and can then learn a by measuring the qubits of the input register.

There is a complementary way to look at this same solution that makes it obvious why it works, without any invocation of massive quantum parallelism. The idea is to note that since U_a functions as a black box, a procedure that does not break open the black box cannot depend on the actual circuitry inside of that box, provided the circuitry does indeed produce the promised output. So we can try to build the black box out of gates that make the effect of the Hadamards on its operation completely transparent.

For the Bernstein-Vazirani oracle this is easily done. We can implement U_a by a collection of cNOT gates, one for each non-zero bit of a. The target of each cNOT is the output register, and the control bits are just those bits of the input register that correspond to the non-zero bits of a. This does precisely what U_a has to do. If the input register starts in the state $|\,x\rangle_n$ and the output register starts in the state $|\,y\rangle_1$, then cNOTs combine to shift y by 1 (modulo 2) for every non-zero bit of x that corresponds to a non-zero bit of a.

With this implementation of the oracle, the Bernstein-Vazirani problem is to find out where the cNOTs that make up U_a have been placed, just by applying U_a, without opening the black box and looking inside it. One cannot solve the problem by opening the black box — the obvious classical move — because the box *might not* in fact, contain this simple collection of cNOT gates. But because U_a acts *as if* it contained such a collection of cNOT gates, if we can learn merely from applying U_a where those cNOT gates would have to be placed to imitate U_a perfectly, we will have learned the value of a whether or not that is how U_a actually does work.

Now consider what happens when $n+1$ Hadamards are applied both before and after applying U_a. If the black box contains cNOT gates, the Hadamards simply convert each of them into a "swapped" cNOT gate that acts the other way around, with control and target exchanged: $(H \otimes H)C_{ij}(H \otimes H) = C_{ji}$. Because we have started with 1 in the *output* register, every one of the swapped cNOT's has 1 for its *control* bit, so it acts as NOT on its target. Because we have started with all qubits of the input register set to 0, and because the *target* qubits of the swapped cNOTs corresponded to the nonzero bits of a, the swapped cNOTS act to set to 1 just those qubits needed to convert the initial state $|\,0\rangle_n$ of the input register into the final state $|\,a\rangle_n$, whose subsequent measurement gives a.

So because U_a acts *as if* it were a collection of cNOT gates controlled by qubits corresponding to the non-zero bits of a, when sandwiched between Hadamards it acts *as if* it were a collection of swapped cNOT gates, whose action, when the state of the input and output registers is initially $|\,0\rangle_n |\,1\rangle_1$, is simply to convert the state of the input register to $|\,a\rangle_n$.

*ndm4@cornell.edu