Evaluation of time and space complexity of a quantum algorithm for the Travelling Salesman problem as compared to its classical computing counterparts for benchmarks.

P vs NP

* P = NP
* Definition of P
  + Deterministic
  + Polynomial Time
* Definition of NP
  + Verifiability in Polynomial Time
  + Reduceability
  + Non-deterministic algorithms
* Big-O Notation

Complexity Theory

* Time Complexity Analysis
* Space Complexity Analysis

Travelling Salesman

* Definition
* Brute-force algorithm
* Current Best-Known Heuristic Algorithm

History of Quantum Computing

* Shor’s Algorithm for Integer Factorization
  + How Shor’s was analyzed for time & space complexity
* Future Projections on qubit availability

Introduction to Notation

* Dirac Notation for Quantum Computing
  + Basis Set
  + Cross Products
  + Direct Products
  + Direct Sums
* Quantum Gates and Circuits
  + Pauli-X
  + Hadamard
  + Controlled
  + Interaction of Hadamard Bases on C-gates
  + Toffoli

Quantum Fourier Transform

Measurement

Srinivasan et al. Brute-Force Algorithm for TSP

Bang et al. Heuristic Algorithm for TSP

INTRODUCTION TO NOTATION

The state of a quantum computer is often represented with Dirac notation. The following ket-vectors: represent the 0 and 1 states (analogous to bits of 0 and 1 in a classical computer) of a single qubit within a quantum computing system. These states can be re-represented as vectors of the forms and , respectively.

To expand the system for multiple qubits, one takes the direct product of all qubits within the system. For example, a two-qubit system, with both qubits in state 0 is expressed by the following:

and in vector format:

.

The associated bra-vector, or co-vector, for each state () is given as the transpose, row-vector, of the state with the complex-conjugate taken for each element. Therefore, we show the following: .

Dirac notation typically omits the operator symbols when operating on two states within an equation. For clarification, we look at the three combinations of two states:

* implies direct product, as shown above;
* implies scalar product, such that: ;
* implies cross product, such that: .

QUBIT STATE

Because a qubit does not exist discretely in the states of or , we must find the probability that the qubit will be measured in either state. The probabilities are given by complex coefficients operating on either state, typically denoted α and β as scalars for the states and , respectively. Thus, we can describe the overall state of a single qubit given:

.

We find the probability amplitudes that a qubit is measured as a 0 or 1 given the probability amplitudes of these coefficients: [1].

QUANTUM GATES

Hadamard Gate

The Hadamard gate acts to superpose the state of a qubit, giving equal probability that the qubit is measured as either 0 or 1 [2].

Hadamard Gate: . When applied to a qubit of either the or basis vectors:

;

.

Likewise, we can refer to the superposed state produced as the Hadamard basis vectors and :

Pauli-X Gate

The Pauli-X gate is the equivalent of a not-gate in classical computers, and acts to invert the values of the and states on a single qubit. This gate is represented by the Pauli-X matrix: .

It is the case then that .

Controlled Gates

Controlled gates require two qubits wherein 1 of the qubits acts as the control, while the other is being acted upon. A Controlled gate is notated as , where is the gate to be controlled [2]. In its generalized form, a controlled gate is constructed as: [3].

The effect is that the controlled gate only acts on the 2nd qubit if the control qubit’s state is 1, and has no effect if the control qubit’s state is 0.

The Controlled-Not gate (CNOT or ) then takes the form:

.

This gate then maps the following two-qubit system as:

With the following as an example: