Quantum Algorithm for Linear Programming Problems

Pramod S. Joag
Department of Physics, University of Pune
PUNE-INDIA
In collaboration with
Dhananjay P. Mehendale

Quantum computation is the evolution of a multi-qubit quantum system from the initially prepared input state to a target output state.

Such evolution is realized by successive application of quantum gates, which are unitary operators acting on one or more qubits.

An N-qubit register can store 2^N numbers in a state which is superposition of 2^N computational basis states.
N = 40 ~2^40~10^12 numbers.
All these 2^N numbers can be processed in parallel by application of N-qubit quantum gate.

However, when we try to read these numbers by measuring the state to the computational basis, the state collapses to one of the basis states, i.e., we can only read one of the 2^N stored numbers at a time. The typical problems where this enormous storage and parallel processing capacity can be used are those involving some global properties of functions whose values are stored in some multi-qubit quantum state. For this class of problems quantum algorithms have the promise of exponentially speeding up the solutions over their classical counterparts.

Quantum algorithms are based on two inherently quantum phenomena: Entanglement & Quantum Interference

Entanglement: Enables the coding of data in a non-trivial superposition of states which cannot be factored into individual qubit states.

Quantum Interference: The key ingredient evolving initial states (input) to final states (output) modifying intermediate multi-qubit superpositions in some prescribed way.

Quantum Algorithms based on Quantum Fourier transform:

1) Phase Estimation
2) Period Finding
3) Prime Factorization

Other Quantum Algorithms:

1) Quantum Search
2) Quantum Simulation

We present a quantum algorithm to solve LP problems, based on the quantum algorithm for a system of linear equations by Seth Lloyd and collabotators.

A. W. Harrow, A. Hassidin, S. Lloyd, PRL, 103, 150502,2009.

We have shown that a general LP problem can be solved using an iterative procedure, where each iteration involves the least squares solution of a system of linear equations which forms a core of the computation. We employ Lloyd's algorithm to solve the system of linear equations in each iteration.

This quantum algorithm exponentially speeds up the execution as well as exponentially reduces the memory requirement
N variable LP Problem requires
$n=O(\log N)$ time steps, &
$n=O(\log N)$ qubit register

LP Problem

Constrained Optimization of a linear expression in N variables, $C^T x$ where c is the vector of profit/cost coefficients. Vector x > = 0 is to be found such that $C^T x$ is either maximum or minimum subject to m constraints involving linear expressions in $\{x_1, x_2, x_3, \square, x_N\}$ in the form of inequalities.

Thus, the constraints are either
Ax <=b or Ax>=b.
These can be turned into equalities by adding m slack variables, $\{s_1, s_2, s_3, \square\ , s_m\}$ or m surplus variables $\{t_1, t_2, t_3, \square\ , t_m\}$.
Thus, the constraint equations become
Ax + s = b,
or,
Ax – t = b.

Therefore, the cannonical forms of LP problems are

Maximize $C^T x$

Subject to Ax + s = b (I)

x>=0, s>=0,

or

Minimize $C^T x$

Subject to Ax - t = b (II)

x>=0, t>=0.

x: Solution vector, c: Profit/cost vector

A: matrix of constraint coefficients,

b: boundaries of constraints.

The Primal LP Problem & its Dual:
Interchanging the roles of b and c we get the dual problem of (I):
Minimize $b^T w$
Subject to A^T w - t = c
w>=0, t>=0.

Duality Theorem: Optimal values of the primal problem and its dual are identical.

$C^T x = b^T w.$

Duality theorem implies:

Solving an LP problem is equivalent to finding a nonnegative solution of a system of equations:

$C^T x - b^T w = 0$

$Ax + s = b$

$A^T w - t = c$

This system can be written as

$$Ez = f$$

where

$$E = \begin{bmatrix} C^T_{1\times n} & 0_{1\times m} & -b^T_{1\times m} & 0_{1\times n} \\ A_{m\times n} & I_{m\times m} & 0_{m\times m} & 0_{m\times n} \\ 0_{n\times n} & 0_{n\times m} & A^T_{n\times m} & -I_{n\times n} \end{bmatrix}$$

$$z = \begin{bmatrix} x \\ s \\ w \\ t \end{bmatrix} \qquad f = \begin{bmatrix} 0_{1\times 1} \\ b_{m\times 1} \\ c_{n\times 1} \end{bmatrix}$$

and where, $\begin{bmatrix} x \\ s \end{bmatrix}$ is solution of

primal problem and $\begin{bmatrix} w \\ t \end{bmatrix}$ is the solution of its dual.

Finding nonnegative solution of this system implies
finding z >=0 which minimizes
$\| Ez - f \|$
this is called Nonnegative Least Squares (NNLS) solution of $Ez = f$

The corresponding algorithm called NNLS algorithm is based on the following theorem called Kuhn-Tucker conditions.
The vector z is a solution of $Ez = f$
subject to Z>=0 iff there exists a vector y and a partitioning of the set {1, 2, …, m}, P, Q, such that

$$y = E^T (Ez - f)$$

$$z_i = 0 \forall i \in P$$

,

$$z_i > 0 \forall i \in Q$$

$$y_i > 0 \forall i \in P$$

$$y_i = 0 \forall i \in Q$$

The corresponding algorithm starts with two sets P = Null set, and Q = {1, 2, …, n} and iteratively constructs the vectors y and z. The crucial step in this algorithm is to solve the system of equations $E_P x = f$ where E_P is a m by n matrix, $x_{n\times 1}$ is the solution and $f_{m\times 1}$ is given.

We can rewrite this equation

$$\begin{bmatrix} 0 & E_P \\ E_P^T & 0 \end{bmatrix} \begin{bmatrix} 0 \\ x \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

Now the matrix acting on the solution vector is a symmetric square matrix, a requirement to implement the quantum algorithm for $E_P x = f$ with E_P, x, and *f* redefined as above.

We seek a least squares solution of $E_P x = f$

i.e. If E_P is invertible

$$x = E_P^{-1} f$$

If E_P is not invertible

$$x = E_P^{-} f$$

where E_P^{-} is the pseudo-inverse of E_P .

Pseudo-inverse

Given a m by n (or noninvertible n by n) matrix A, its pseudo-inverse can be constructed as follows. Write

$$A = USV^{+}$$

$$S = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{where } \Sigma = diag(\sigma_1, \square\ , \sigma_r)$$

$$S^{-} = \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} \quad \text{where } \Sigma^{-1} = diag(\sigma^{-1}{}_1, \square\ , \sigma^{-1}{}_r)$$

For every m by n matrix A and every n by 1 vector b the vector $x = A^{-}b$ is the least squares solution of $Ax = b$

Quantum Algorithm:

The given system of equations is encoded as $A|x>=|b>$

We wish to find $|x>=A^{-1}|b>$ (or $|x>=A^{-}|b>$).

We evaluate

$$\exp(-iA\Delta t \otimes p)|b>|0>=\sum_j \beta_j |\psi_j>|\lambda_j \Delta t>$$

Using phase estimation technique we can transform this to $\sum_j \beta_j \exp(i\Delta t\lambda_j^{-1}|\psi_j>|\lambda_j \Delta t>$

By de-phasing we get

$$\sum_j \beta_j \exp(i\Delta t \lambda_j^{-1} |\psi_j\rangle |0\rangle = \exp(i\Delta t A^{-1}) |b\rangle |0\rangle = (1 + i\Delta t A^{-1}) |b\rangle |0\rangle$$

From this we can easily extract the required

$$|x\rangle = A^{-1} |b\rangle$$

or

$$x = A^{-} |b\rangle$$