BOSCH's
Route Optimization Algorithm

Problem Statement

Given n nodes, we need to find an optimal path that satisfies a set of constraints.

Vehicle occupancy should be at least 85%

Fixed size clustering ensures cluster size to be limited in a range

Minimize operational cost.

The path planning algorithm is designed to minimize cost

Number of buses

Minimum number of clusters are formed assigning one bus to each cluster

Time window for pick up and drop off

Total duration for the bus is limited by the time windows

Problem Statement

Given n nodes, we need to find an optimal path that satisfies a set of constraints.

Vehicle occupancy should be at least 85%	Minimize operational cost.	Number of buses	Time window for pick up and drop off
Fixed size clustering ensures cluster size to be limited in a range	The path planning algorithm is designed to minimize cost	Minimum number of clusters are formed assigning one bus to each cluster	Total duration for the bus is limited by the time windows

Problem Statement

Given n nodes, we need to find an optimal path that satisfies a set of constraints.

Vehicle occupancy should be at least 85%	Fixed size clustering ensures cluster size to be limited in a range
Minimize operational cost.	The path planning algorithm is designed to minimize cost
Number of buses	Minimum number of clusters are formed assigning one bus to each cluster
Time window for pick up and drop off	Total duration for the bus is limited by the time windows

Problem Statement

Given n nodes, we need to find an optimal path that satisfies a set of constraints.

Vehicle occupancy should be at least 85%	Minimize operational cost.	Number of buses	Time window for pick up and drop off
Fixed size clustering ensures cluster size to be limited in a range	The path planning algorithm is designed to minimize cost	Minimum number of clusters are formed assigning one bus to each cluster	Total duration for the bus is limited by the time windows

Why not use Machine Learning?

In a very recent paper (Sep 2019), Francois et al. conducted a comprehensive study showing that despite the apparently good results of learning approaches, the performances are still far from the traditional approaches used to solve TSP like LKH algorithm and Christofides algorithm.

Approach - Approximation Algorithms

1

Removing constraints & Min-Cost Path

We started by removing the constraints given in the original problem statement and solving the problem of visiting n stops with a min-cost path.

2

Literature Survey of Travelling Salesman

It can be easily shown that the given problem is a special instance of a Travelling Salesperson Problem(TSP) and thus we started with an extensive literature survey of TSP.

3

NP-hard (Karp 1972)

As, the TSP problem is proven to be NP-hard(Karp 1972), we approached solving it via Approximation algorithms and Heuristic algorithms.

4

Christofides Algorithm

As of 2019, the best known approximation ratio is provided by **Christofides algorithm** which solves it with an approximation constant of 1.5

5

Testing on datasets

Although,christofides algorithm ensures an upper bound by a constant factor, by analysing several route optimisation datasets online, we found that on average, it performs worse than most SOTA heuristic algorithms.

Approach - Heuristic Algorithms

1

Lin-Kernighan Heuristic algorithm

We found Lin-Kernighan heuristic to be the most promising as it has produced optimal solutions for a 109399-city instance (which to the best of our knowledge is the largest nontrivial instance solved to optimality).

2

Dynamic programming based heuristic algorithm

Taking inspiration from the LKH algorithm, we propose a dynamic programming based heuristic algorithm to solve the given challenge.

3

DP solution matrix

We construct a 3 dimensional array named Path such that Path[i][j] stores a vector of indices denoting the min-cost path visiting "i" number of vertices and ending at the specific vertex "j"

4

Update step

We start filling this array in a bottom-up approach where in each computation step we try to not repeat any vertices that have already been occured in the path.

5

Time Complexity

Finally, the algorithm outputs the optimal path among all such paths covering N vertices. The time complexity of our proposed approach is n^3 where n denotes the number of stops taken by the bus.

Approach - Adding capacity constraints

Capacity constrained clustering

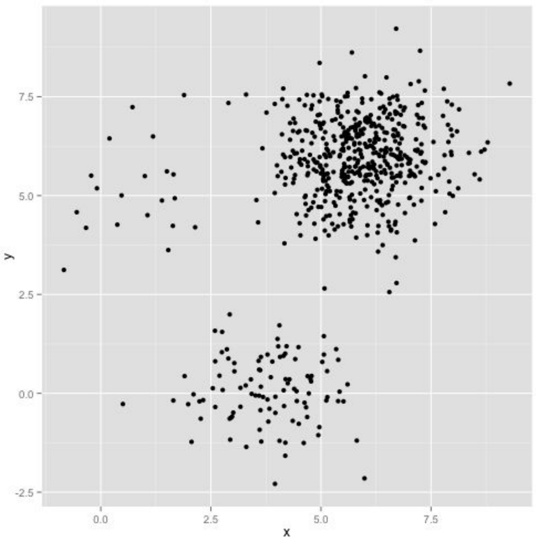
To satisfy the capacity constraint laid out in the problem statement, we proceed by clustering the points around the depot

Why not K-means

We proceeded by first trying out K-means on the points to cluster them but ended up rejecting the approach as K-means often resulted in uneven cluster formation which grossly violated the capacity constraints

Line Sweeping Algorithm

To ensure even clusters, we used polar-coordinate representation of points and line-sweeping algorithm which resulted in much better cluster formations and also ended up enhancing the optimality of various instances of TSP problem



Approach

1 **User Interface**

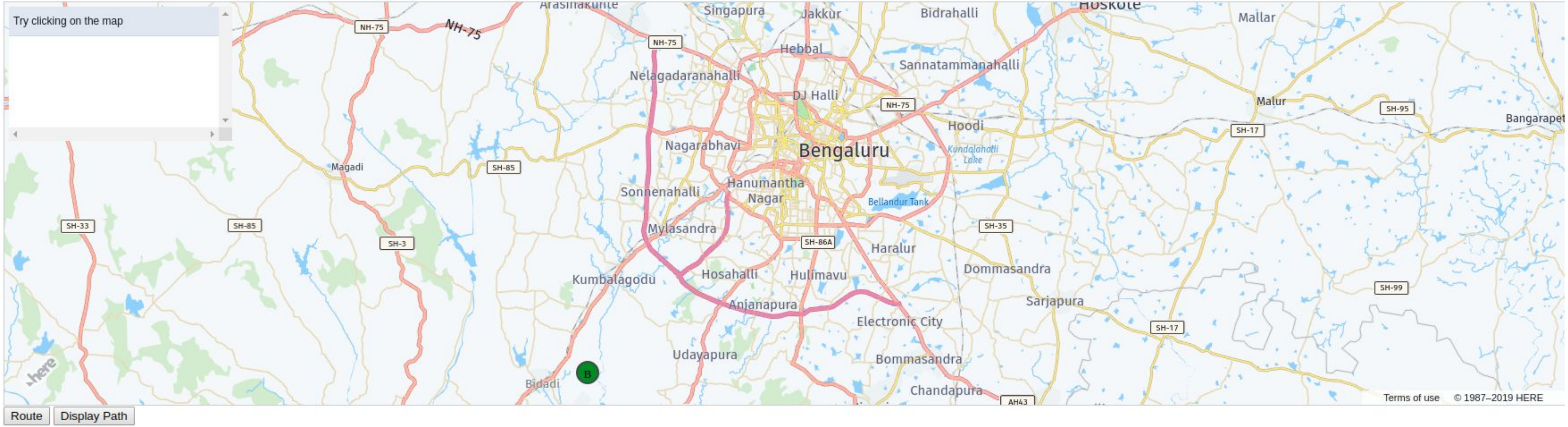
Markers are placed on the map, with their destination on Bosch Bidadi. A dedicated mobile app is developed for the passenger to extract and mark present location

2 **Clustering**

The points marked are clustered to satisfy the capacity constraint, and one bus caters each cluster.

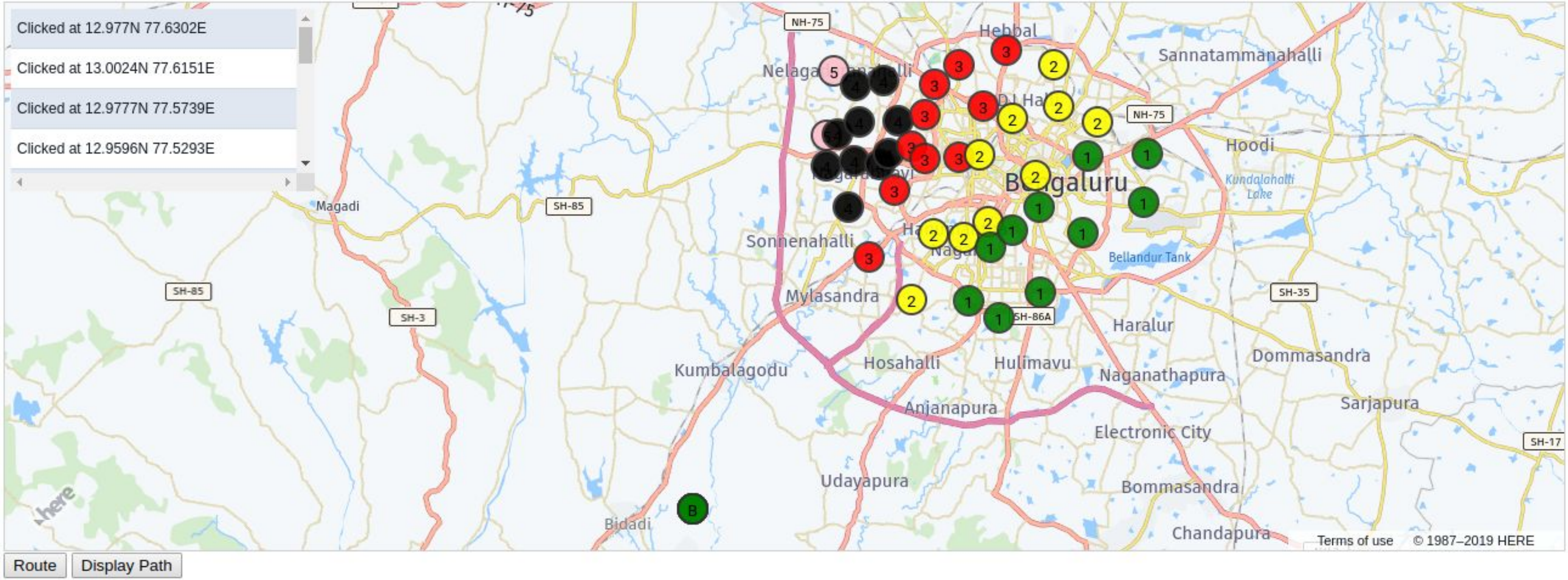
3 **Path planning**

Optimal path in a cluster formed for each bus to follow satisfying the constraints.

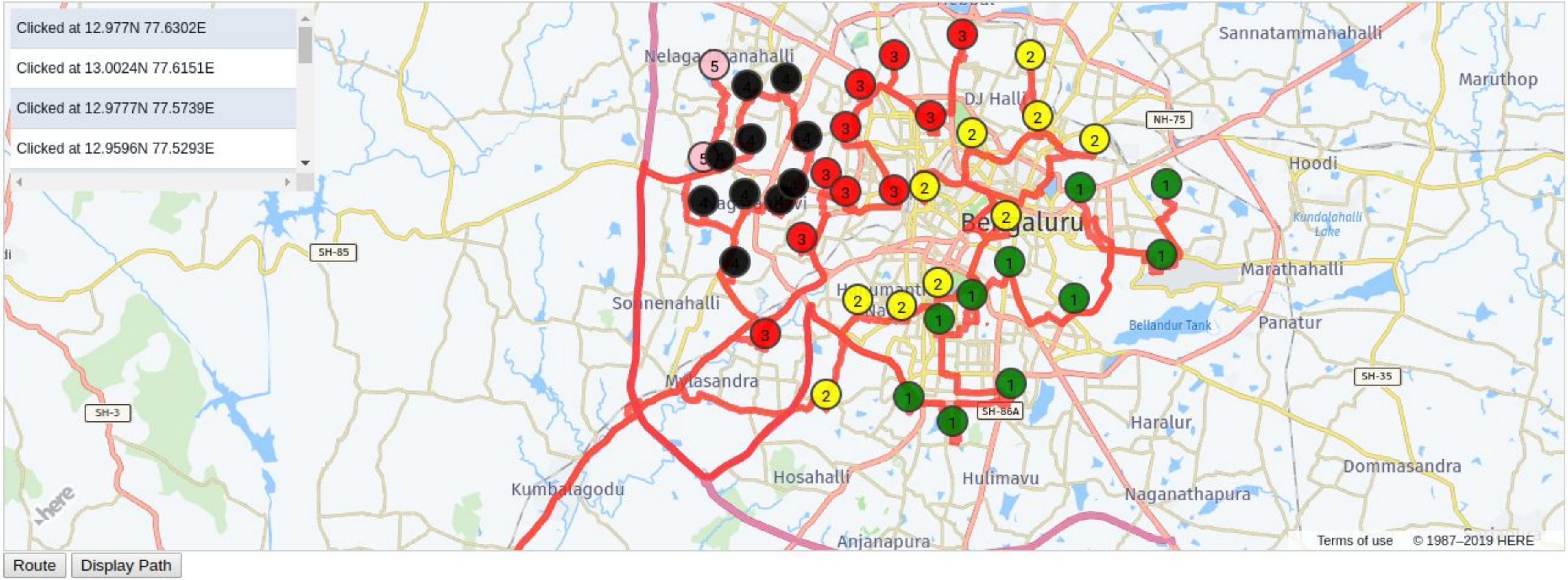


Try clicking on the map
Route
Display Path
Bengaluru
Singapura
Jakkur
Bidrahalli
Mallar
Hebbal
Sannatammanahalli
Nelagadaranahalli
DJ Halli
Hoodi
Malur
Bangarapet
Nagarabhavi
Kundalahalli Lake
Magadi
Sonnenahalli
Hanumantha Nagar
Bellandur Tank
Mylasandra
Haralur
Dommasandra
Kumbalagodu
Hosahalli
Hulimavu
Sarjapura
Anjanapura
Electronic City
Udayapura
Bommasandra
Bidadi
Chandapura
NH-75
SH-85
SH-33
SH-3
SH-86A
SH-35
SH-17
SH-95
SH-99
AH43
B
here
Terms of use
© 1987–2019 HERE

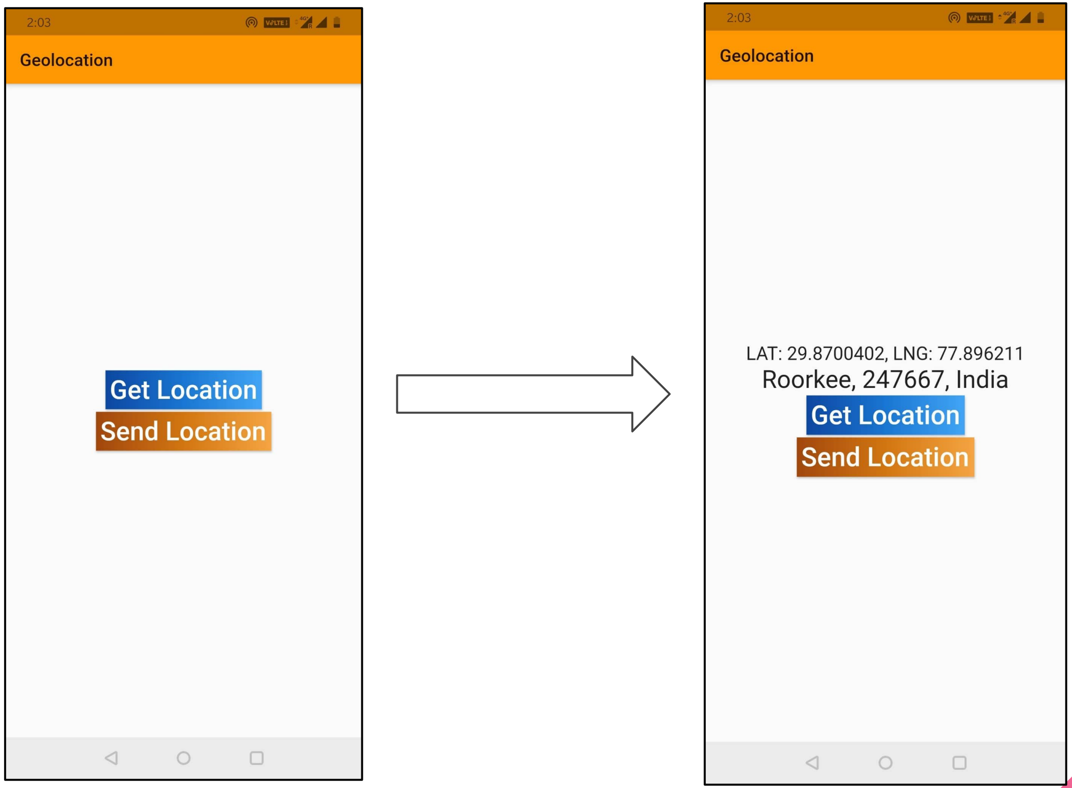
Route Display Path



Clicked at 12.977N 77.6302E
Clicked at 13.0024N 77.6151E
Clicked at 12.9777N 77.5739E
Clicked at 12.9596N 77.5293E
NH-75
Hebbal
Sannatammanahalli
NH-75
Hoodi
Kundalahalli Lake
Magadi
SH-85
Sonnenahalli
Bellandur Tank
SH-85
SH-3
Mylasandra
SH-86A
SH-35
Haralur
Dommasandra
Kumbalagodu
Hosahalli
Hulimavu
Naganathapura
Sarjapura
Anjanapura
Electronic City
SH-17
Udayapura
Bommasandra
Bidadi
Chandapura
Terms of use
© 1987–2019 HERE
Route
Display Path


Clicked at 12.977N 77.6302E
Clicked at 13.0024N 77.6151E
Clicked at 12.9777N 77.5739E
Clicked at 12.9596N 77.5293E
Sannatammanahalli
Maruthop
DJ Halli
NH-75
Hoodi
Kundalahalli Lake
Bengaluru
Marathahalli
Sonnenahalli
Bellandur Tank
Panatur
SH-85
SH-35
SH-3
SH-86A
Haralur
Hosahalli
Hulimavu
Dommasandra
Naganathapura
Kumbalagodu
Anjanapura
Terms of use
© 1987–2019 HERE
Route
Display Path

Android Application for passengers



Challenges

For the given optimization challenge, mathematically, more than a trillion possible combinations exist for scheduling pickup and drop of even 15 people in adjacent time slots.But which schedule represents the optimal solution given dynamically changing variables such as timing constraints of passengers, potential delays due to road obstructions and the operational costs of transport?

Even when applying approximation and heuristic techniques, the number of possibilities is still far too large for a classical computer to explore **on the fly**.

The Quantum Revolution

"When you change the way you look at things, the things you look at change"

- **Max Planck**, Father of quantum physics

What is Quantum Computing?

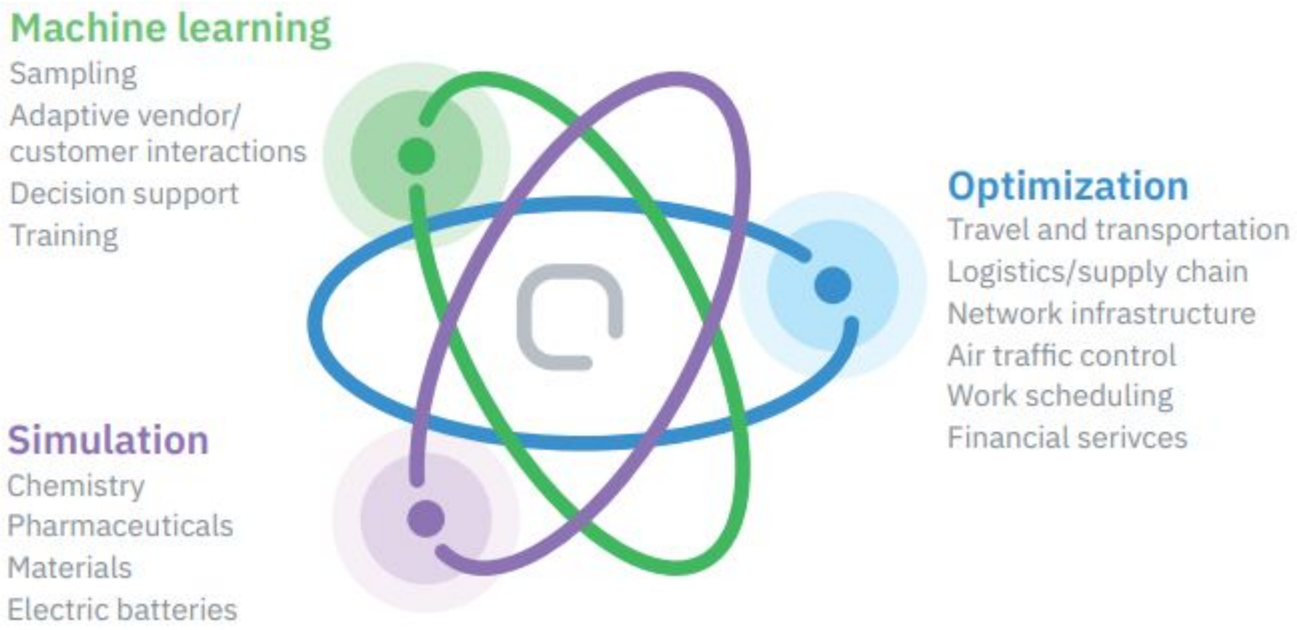
Quantum computing takes advantage of the laws of quantum mechanics found in nature and represents a fundamental change from classical information processing. Two properties of quantum behavior – **superposition** and **entanglement** – allow quantum computers to solve problems intractable for today's conventional, or classical, machines

The Potential

Quantum computing's potential for significant speedup over classical computers[1]

Type of scaling	Time to solve problem				
Classical algorithm with exponential runtime	10 secs	2 mins	330 years	3300 years	Age of the universe
Quantum algorithm with polynomial runtime	1 min	2 mins	10 mins	11 mins	~24 mins

Near-term quantum applications



Quantum Approximate Optimization Algorithm

To tackle the dynamic nature of constraints, we propose a Quantum Approximation Optimization Algorithm (QAOA) based approach.

We implemented the Traveling Salesman Problem (TSP) using the QAOA on Rigetti's QVM.

A Quantum Approximate Optimization Algorithm

Edward Farhi and Jeffrey Goldstone

Center for Theoretical Physics

Massachusetts Institute of Technology

Cambridge, MA 02139

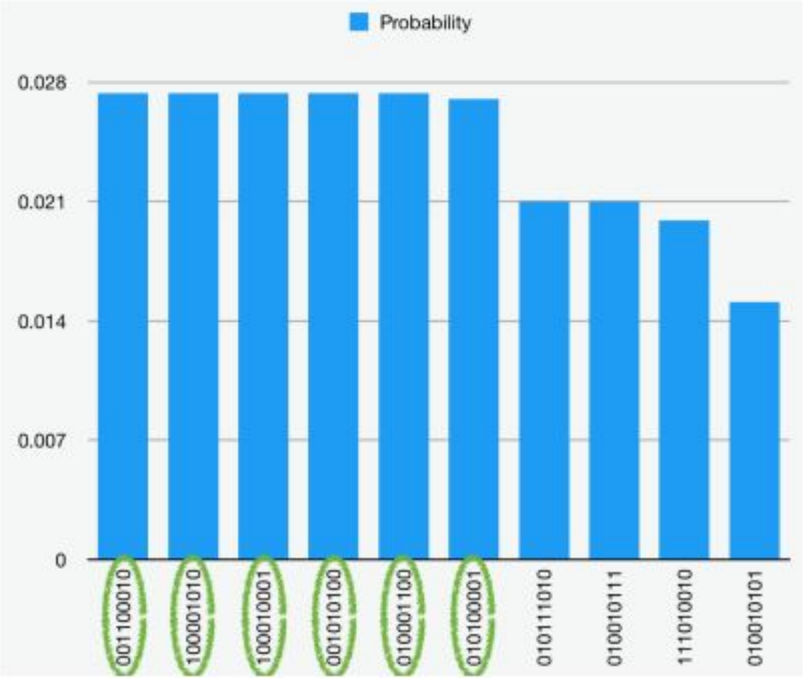
Sam Gutmann

Abstract

We introduce a quantum algorithm that produces approximate solutions for combinatorial optimization problems. The algorithm depends on an integer $p \geq 1$ and the quality of the approximation improves as p is increased. The quantum circuit that implements the algorithm consists of unitary gates whose locality is at most the locality of the objective function whose optimum is sought. The depth of the circuit grows linearly with p times (at worst) the number of constraints. If p is fixed, that is, independent of the input size, the algorithm makes use of efficient classical preprocessing. If p grows with the input size a different strategy is proposed. We study the algorithm as applied to MaxCut on regular graphs and analyze its performance on 2-regular and 3-regular graphs for fixed p. For $p = 1$, on 3-regular graphs the quantum algorithm always finds a cut that is at least 0.6924 times the size of the optimal cut.

arXiv:1411.4028

Quantum Approximate Optimization Algorithm



Our TSP QAOA implementation is successful at finding the classical solution for graphs with cities up to 4, but requires at least about 1000 circuit output samples for consistent results.

References

1. Edward Farhi, Jeffrey Goldstone, Sam Gutmann. A Quantum Approximate Optimization Algorithm. 2014. URL https://arxiv.org/abs/1411.4028

2. Stuart Hadfield, Zhihui Wang, Bryan O’Gorman, Eleanor G. Rieffel, Davide Venturelli, Rupak Biswas. From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. 2017. https://arxiv.org/abs/1709.03489

3. Lou K. Grover. A fast quantum mechanical algorithm for database search. 1996. URL https://arxiv.org/abs/quant-ph/9605043

4. Karthik Srinivasan, Saipriya Satyajit, Bikash K. Behera, Prasanta K. Panigrahi. Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience. 2018. URL https://arxiv.org/abs/1805.10928

5. Dominic J. Moylett, Noah Linden, Ashley Montanaro. Quantum speedup of the Travelling Salesman Problem for bounded-degree graphs. URL https://arxiv.org/abs/1612.06203

6. https://github.com/danielhenry1/QAOA_TSP

THANK YOU