White paper for OGC (add title text)

NOTE	*Preface* Insert Preface Text here. Give OGC specific commentary: describe the technical content, reason for document, history of the document and precursors, and plans for future work.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Abstract

<Insert Abstract Text here>

Introduction

This discussion paper is concerned with the topic of quantum computing (QC) and associated technologies and its applicability to addressing geospatial problems. The Open Geospatial Consortium (OGC) is an innovative, member led organization concerned with creation and maintenance of standards for geospatial data, processing, and adjacent technologies. It has been operating for over 20 years with a guiding principle that geospatial assets should be FAIR (Findable, Accessible, Interoperable, Reusable). Quantum computing is of interest to the OGC because of its applicability to uniquely complex, geospatial problems.

The focus of the experimentation described in later sections was *integration* and *experimentation* rather than discerning new quantum algorithms or looking at generation after next capabilities. Part of the motivation was the pace of technological change and specifically the rise of technologies such as ChatGPT and LLMs in general which came to public prominence almost overnight and answering the following questions:

- What if QC has it's *ChatGPT moment?*
- What could QC be used for in the geospatial domain?
- What is the integration path for QC with existing geospatial technologies?

Although quantum technologies are not new, the recent interest QC for computationally difficult applications is notable with many large technology companies spinning up quantum divisions in both the applications and engineering specialisms. Additionally, there are specialist companies working on engineering QCs, with further private entities working on applications. This paper provides a discussion on applications and potential applications for quantum computing, although

there is an introduction to some of the concepts and challenges with QC it is largely focused on the practical applications of QC for geospatial problems but does provide a cursory discussion of the approaches to QC and the challenges with the engineering and hardware. As with many technologies, QC is constantly evolving in terms of capabilities, speed and overall hardware.

The promise of QC has been understood prior to any hardware implementations. Perhaps the most well-known (and for some, concerning) applications is the ability for QCs to factor very large numbers. Although this may seem a trivial application for these machines, many encryption protocols are built upon the principle that deriving prime factors from a very large number is computationally hard, although not impossible. A salient example is the use of public and private keys for securing pieces of information where both the public and private key are required to decrypt information and read the contents. Public key are transmitted with the understanding that they maybe intercepted, but because of the nature of computing factors from large numbers using classical computing, it is computationally hard to derive the private key with knowledge of the public key. However, generating the public key from the private key is computationally trivial. Although current QCs are not able to crack this type of encryption with current, small to medium sized machines, encrypted information can be collected with the ambition of decrypting it in the near future. Interestingly, the ability for QC to factor large numbers was proven in the 1990s by Peter Shor (Shor's algorithm) long before practical implementations of QCs.

Another often cited QC application is search of large databases. This application was investigated by Grover who produced an algorithm. As with Shor's algorithm, a quantum advantage to searching a database was shown albeit quadratic rather than exponential. An additional consideration for practical implementation is the reliance on an *oracle function*, which somehow derives what the result of the search should be and the QC locates the record or records in the database.

There are several other applications that are gaining traction for QC use cases including interaction with sensors and *quantum machine learning*. At time of writing, the ML applications do not have the same level of *proof* as the previous examples, but have shown promise.

Quantum Computing

Classical computers work using *bits*. Bits are the fundamental building blocks of computation that can have one of two states and any one time (usually styled as 0 and 1 or *off* and *on*). QC is different in that its building blocks are *qubits*, which exist in a superposition of 0 and 1. Often this is visualized as *Bloch Sphere* where a spin upwards is analogous to *0* and a spin downwards is *1*. Any other spin is a superposition of 0 and 1 because it contains components of both. This ability to hold a superposition of qubits is fundamental to QC approaches.

A second component of QC is *entanglement*, which was famously dismissed by Einstein as "spooky action at a distance." Entanglement happens when two particles are linked and can effect each other's state regardless of their distance from each other. This also has implications for quantum teleportation and information exchange, although this does not cause issues with causality because information regarding the state change on one particle is required to get information out of the second particle and the transmission of the information must be done classically with all of the speed of light restrictions associated with it.

An additional consideration of QC (of which there are many) is that the process is non-

deterministic, unlike classical computing that will produce the same result every time provided the initial conditions are identical, QC can produce different results with the same input. A method of addressing this issue it to perform the calculation many times (1000s) and averaging the results in some way (usually the mode).

Types of quantum computing

QC implementations have emerged from large technology companies as well as small and middle size disruptive businesses looking to leverage practical QC as the technology evolves. Unlike *classical computing* which is generalized, QC is specialized to solve certain types of problems. Although QC shows promise for some applications, it is not currently envisaged that QC or Quantum Processing Units (QPUs) will replace classical computers, this is not like the invention of the microchip replacing the valve approach to generalized computation! Currently there are two conceptual approaches to QC, these are:

- Circuit based.
- Adiabatic or annealers.

Circuit based QC are potentially closer to generalized classical computing. These are the machines that can factor large numbers and perform search algorithms as described in the introduction. Indicatively, this type of QC is where much of the focus of the research community lies, this maybe because it has the potential to offer quantum advantage, that is, it can solve problems that cannot be solved in a sensible time by a classical computation approach (note that this includes the world's most powerful supercomputers).

Adiabatic quantum computing (AQC) is a different type of computation that is setup specifically for solving optimization problems. The hardware and topology of AQC is completely different from circuit based approaches, which means that unlike generalized classical computing, you cannot run circuit based algorithms on AQCs as they way that they perform computation is fundamentally different. Simplistically, whilst circuit based QC is concerned with the manipulation and transformation of individual qubits, ADQ looks to put the entire system in a very low energy state and then transfer the problem space slowly into the AQC with the result to the entire problem emerging from maintenance of the low energy state.

There are many online resources that provide a fuller explanation including the physics of QC and quantum information theory, but it is beyond the scope of this discussion paper that is focused on applications.

Quantum hardware and engineering challenges

The hardware and engineering challenges with QC have contributed to its relatively slow progress compared to other technologies such as artificial intelligence, or the increase in power of graphical processing units (GPUs). Two big challenges with QC hardware are *noise* and *decoherence*. Qubits are fragile and cannot hold state for long periods of time. There is also the fundamental problem of interacting with a system that for all intents and purposes needs to be isolated from the outside work to maintain coherence. One method of addresses these issues is to increase the number of physical qubits to create a smaller number of *logical* qubits where the outputs from each bank of physical qubits is corrected. The result of this approach is that it makes quantum computation far

more reliable that without the correction, however, it requires many times more physical qubits to produce a number of logical qubits that can do computation of practical and useful size.

From a hardware perspective, there are different approaches to creating physical machines and addressing the challenges of noise and decoherence. Some examples are:

1. Superconducting approaches - where materials are cooled to close to absolute zero to produce quantum effects.
2. Trapped ions - ions trapped using electromagnetic fields.
3. Photons - using particles of light to perform computation.
4. Neutral atoms - use of lasers to arrange atoms into grids.
5. Annealers - use of phenomenon *quantum tunnelling* to find the minima of a function for optimization problems (the main concern of this paper with respect to geospatial applications).

There are other methods being actively investigated, however, the number of approaches demonstrates that experimentation is still being conducted, and the *best* technology has yet to be settled on. This is in contrast to classical computing where silicon and to a lesser extent copper are the metals used to create microchips.

Overall, these technologies represent engineering challenges. Unlike CPUs which reside locally inside classical computers, it appears unlikely that a QPU will sit along side CPUs like GPUs do. The current generation of QCs are of a similar size to the early classical mainframes and therefore require infrastructure to host locally. The main method of accessing quantum computation today is to use cloud connected services.

Interaction with quantum computers

Creating and executing jobs for todays quantum computers is done through software development kits (SDKs). As QPU time can be expensive, the SDKs often offer a *quantum simulator* to test code locally with the quantum output simulated via a classical computer. Some SDKs, especially for circuit based machines offer the ability to create and simulate quantum circuits. Individual qubits are tasked within the circuit to produce a result; although this is useable for the size of QCs that are available now, it will quickly become impractical with even a small number of qubits. Many of these SDKs offer the ability to call a QC with parameters to execute a known computation without having to do low-level programming, a salient example of this is with quantum annealing where the problem space is restricted to optimization problems. Some examples of SDKs are as follows:

1. Qiskit - IBM - used for constructing and executing circuits.
2. Cirq - Google - Python library tailored to Google's quantum machines.
3. Ocean Software - used for quantum annealing on D-Wave's machines.
4. Microsoft Quantum Development Kit - uses Q#, a language for expressing quantum algorithms.

There are also many more SDKs with some intending to be cross platform.

Much of the work described in this paper was conducted using the Ocean Software SDK to execute D-Wave cloud connected quantum computers.

Geospatial Applications

The Open Geospatial Consortium (OGC) is a longstanding and premier standards body for the geospatial industry. With over 20 years of history, the OGC has been a thought leader in emerging technologies and whose standards have facilitated interoperability across IT enterprises and seen wide adoption in industry leading products and governmental organizations.

As discussed in previous sections, the geospatial domain has many problems that maybe addressable by QC. For example, geospatial data has always been *large*, long before the concept of *big data* and finding items with fast search algorithms has been a topic of interest for some time, especially when spatial operators are involved which increases the complexity. Beyond search is the concept of *optimization*, which is broadly the solving of complex computations to find the *best* solution given a very large number of potential solutions. From a geospatial perspective this includes the following:

1. Routing - given the possibility of many routes, which is the optimal?
2. Optimal location planning - given a list of possible locations, where should I place my services to give the largest number of people access within a time constraint?
3. Geospatial relationships - given a list of locations, people, and relationships between them what patterns can be discerned?
4. Knowledge graphs - given a number of entities and relationships, what is the optimal path to get from a to be given c and d?
5. Orbital mechanics and communications - what is the optimal constellation of satellites given these constraints?

Optimization problems are often transferrable to other, adjacent problems, and because many of these problems are well known and reducible to other problems, the programming required to express them in quantum terms is often reduced to a call to a server with parameters rather than starting from first principles.

The following sections outline the general approach to AQC for solving optimization problems and some practical use cases that have been tested by integration of AQC into current geospatial software capabilities.

Computational difficulty of optimization problems

Not all optimization problems are the same but they do have some commonalities, the main one being that there are many solutions to the problems stated and finding the correct (optimal) one is the challenge because of the number possible solutions available. Many optimization problems have a factorial (noted as n!) component within them. This means that for every layer of complexity, node, destination, relationship, or variable added to the problem makes finding the solution exponentially more complex. An often cited example is the number of possible calculations in a standard deck of 52 cards, this is clearly 52!. The result of this calculation is 8.1x10^67 possible combinations of a deck of 52 cards, the result of this is that every single deck of cards that has every

existed and will ever exist is likely a unique ordering of the deck (provided that it's shuffled correctly).

The *deck of cards* type problem shows that even with a relatively small number of nodes problems can quickly become unfeasible for the classical computing approach.

Optimization approaches

Binary Quadratic Models (BQMs) are a general method of mathematically expressing optimization problems, the D-Wave system uses these extensively. The quantum approach to using BQMs is to express variables as a binary (0 or 1) and then focuses on optimizing quadratic functions of these variables. A BQM represents the energy state of a system with the QC tasked with minimizing or maximizing the energy state depending on the problem expressed. Like a BQM, which can manage constrained and unconstrained variables, a Quadratic Unconstrained Binary Optimization (QUBO) maybe used when the problem space is open, however, the same principle applies with the AQC tasked with minimizing or maximizing the energy state to produce the solution, which is represented by the state of the machine when the optimization is completed.

Use Cases

This section outlines three geospatially relevant optimization problems that have been tested using D-Wave's quantum annealer. Each use case contains a description of the problem, the approach to sending the problem to the AQC, the results returned and the issues encountered. The AQC was cloud based and could be executed provided the local machine had internet connectivity. For the first two use cases (TSP, SIB) plugin for Esri ArcGIS Pro were created to ensure integration of QC within a geospatial workflow. Additionally, each of the use cases described were created from examples on the D-Wave website as a starting point for the integration exercises.

<website>

All of the problems stated are *at least* NP hard (from the famous P=NP consideration), with Traveling Salesperson Problem potentially being NP Complete according to some sources. This paper does not discuss the implications for NP completeness apart from to provide evidence that the optimization problems discussed are indeed *difficult* for classical computers.

<sources in here>

Traveling Salesperson Problem

The Traveling Salesperson Problem (TSP) is typical in geospatial and logistics. There are different formulations, the one used in this experiment is as follows:

1. There is a salesperson with a known start locations.
2. There are a number of locations that the salesperson must visit.
3. The solution is reached when the salesperson visits each of the locations once and only once in the lowest *cost* possible.

There are several ways to generate *cost* and may include:

- Straight line distance.
- Drive time.
- Road network distance.
- Efficient fuel usage.

Considering a *symmetrical* cost matrix, that is, point A to point B costs the same to travel between as point B to point A, the mathematical formulation of the problem is as follows:

(n-1)!/2

An asymmetrical cost matrix would have an even larger problem space as it would not have the divide by 2 in the formula. The result of this formula is that:

- A 5 city problem has 12 possible routes.
- A 10 city problem has 181,440 possible routes.
- A 100 city problem has 4.7x10^155 possible routes.

As described, the number of possible routes quickly becomes unmanageable for classical computers to solve. Additionally this number of destinations maybe a problem with applicability in the real-world (considering large logistical operations such as Amazon).

A caveat to this problem is that the comparison done is between the brute force version of the classical TSP algorithm (i.e., trying all of the combinations and picking the least cost). In logistics, there are many methods of solving this algorithm that are more efficient such as place aggregation, use of heuristics or other methods of reducing the problem to a manageable size. Therefore the overall utility of AQC in this space is left to the logistics domain experts to judge.

TSP Algorithm preparation

As TSP is a typical geospatial problem, D-Wave has a call that can be made to a quantum machine to solve TSP with the following parameters:

- A cost matrix.
- Configuration of the AQC (number of times to do the computation).

As part of the experiment two cost matrices were generated, one using straight line distance and one using the road network.

TSP Results

The AQC was able to quickly compute the correct route for 5 cities but often failed with 8 cities. As the ADQ is non-deterministic and has some noise in the system, it occasionally produces invalid results such as visiting the same destination twice. When compared to the classical method of doing TSP, the quantum approach is orders of magnitude quicker showing promise for the technology in this space.

Structural Imbalance within a Graph

The Structural Imbalance Problem (SIP) is a special case of the *maximum cut* problem, which is a method of classifying graphs into two groups of vertices where the optimal solution is the one that maximizes the number of edges between the two groups. SIP seeks to classify graphs (usually social networks) using the rule "the enemy of my friend is my enemy".

The mathematical formulation of SIP is as follows:

Minimize ∑ (i,j)□E-wij xi xj

This involves minimizing the objective function according to the positive (friendly) and negative (adversarial) relationships between the entities in the graph. The algorithm can result in two sets of results:

1. A perfectly balanced graph is one where all relationships between individuals within groups are friendly, and all relationships between groups are hostile.
2. An unbalanced graph is one where there are relationships within the graph that break the rule, that is, there are relationships that are friendly that should be hostile and hostile relationships that should be friendly.

The relationships that break the friendly/hostile rules are considered *frustrated*.

A simple Shakespearean Example

A typical example for this type of problem is Romeo & Juliet. At the beginning of the play, the characters sit in a perfectly balanced graph, all of the Montague and Capulet families have positive relationships within their families, and all relationships between the individuals of the two families are negative. As the story progresses, a frustrated relationship emerges with the title characters. If the relationship between the title characters is updated to reflect their positive interactions and the algorithm is re-run, the relationship is flagged as being frustrated, the two characters are in a friendly relationship when they *should* be hostile. This matters because these frustrated relationships can be a predictor of conflict as they are in this story, but also in real life examples.

A real world geospatial example

Applying SIP to geospatial use cases requires:

1. A geospatially enabled dataset.
2. A knowledge graph technology that can handle geospatial operators.

An experiment was carried out using some world terrorist incident data, again provided by D-Wave. As with the TSP example, the objective of this piece of work was to integrate AQC and SIP with geospatial technologies to take advantage of geospatial intelligence with AQC to make some observations about patterns found in the data. A note on the parameters for the data is that the dataset was considered as a whole and not split regionally, which is something that maybe done in a real world scenario. Additionally, due to the large number of data points, the problem set is too large for the QPU alone and a *hybrid solver* was used instead. Hybrid solvers as the name suggests use a combination of classical and quantum technologies to solve larger problems than a pure QPU

could do alone. Exactly how the problem is split up is not described in detail, however it does appear to produce timely and accurate results.

<image in here>

Although the grouping created by the AQC are arbitrary, as the dataset has an unconsidered temporal element, it does highlight areas of the world with many frustrated relationships. The Middle East region is particularly challenging with many frustrated relationships that can be a source of conflict.

Space examples

As optimization can be applied to typical geospatial operations, they can also be applied to any domain where there is a problem with many correct solutions but one being optimal.

Organization, monitoring and controlling satellites whilst taking note of other orbital objects such as space debris contains many optimization problems. Conceptually there are many ways to configure a constellation to achieve certain goals, but there is an optimal solution. The complexity and number of permutations that bodies can be organized implies a proper solution.

Satellite communications optimization

A sample optimization problem was run to ensure efficiency of communications between satellites in a constellation. There are two experiments that were run:

1. All satellites in the constellation must have non-interfering links (no two satellites can share a link)
2. No satellite in the constellation can have more than one adjacent link, to do so assumes interference.

<needs finishing> <needs images>

Satellite constellation placement optimization

Satellite usage and placement in a constellation is an interesting and geospatially adjacent issue what AQC can help with. Although it is inherently a geospatial issue with respect to observing a patch of the earth, the problem can be simplified into a QUBO as mentioned previously. The role of geospatial technologies and data is to provide intelligence into the input data. The example shown here does not use information about satellite orbits and periods, it simply assumes that a constellation of satellites can observe a location at a given time. Whether a satellite can observe a location is represented as simple binary, 1 for it can observe and 0 it cannot observe the location. Additionally, the temporal element is considered as slices, the matrix provides 5 time slices and the binary describes whether the satellite can view the location.

Satellite	Time Slice 1	Time Slice 2	Time Slice 3	Time Slice 4	Time Slice 5
0	1	0	1	0	1
1	0	1	0	1	0

2	1	1	0	1	1

Potential Standardization Routes

Discussion

Recommendations

Quantum is part of a geospatial workflow

It has potential and should be explored for geo

Make use of the hybrid approach and see what that does.

Appropriate use of the technology

Conclusions

Section header

Annex A: Title

NOTE	Place other Annex material in sequential annexes beginning with "A" and leave final two annexes for the Revision History and Bibliography

Annex B: Revision History

Date	Release	Editor	Primary clauses modified	Description
2016-04-28	0.1	G. Editor	all	initial version

Bibliography

NOTE

Example Bibliography (Delete this note).

The TC has approved Springer LNCS as the official document citation type.

Springer LNCS is widely used in technical and computer science journals and other publications

- For citations in the text please use square brackets and consecutive numbers: [1], [2], [3]

– Actual References:

[n] Journal: Author Surname, A.: Title. Publication Title. Volume number, Issue number, Pages Used (Year Published)

[n] Web: Author Surname, A.: Title, http://Website-Url

- [OGCTB12], *OGC: OGC Testbed 12 Annex B: Architecture* (2015).