

Procedural generation with a quantum computer

James R. Wootton
IBM Research - Zurich
Switzerland
jwo@zurich.ibm.com

Abstract—

Index Terms—component, formatting, style, styling, insert

I. INTRODUCTION

Quantum computers offer a radically new form of hardware, with radically new software to go with in. Over the last few decades, many algorithms have been developed which offer a provable reduction in computational complexity in comparison with conventional digital computing. In some cases this is a polynomial speed-up (such as a reduction from $O(n)$ to $O(\sqrt{n})$ for search problems). In other cases the speed-up is super-polynomial or exponential (such as the polynomial complexity quantum algorithm for factoring). These algorithms address a variety of different types of problem, including optimization problems and graph-theoretic analysis.

The hardware to run such algorithms is still very much in a prototype stage. Until now, hardware limitations have meant that it is far easier to simulate quantum devices on a standard laptop than to actually use the real versions. It is only with the current generation of 53 qubit devices from IBM and Google that quantum hardware can only be matched by days of dedicated usage of the world's largest supercomputer. Even so, the implementation of such 'textbook' algorithms is still some years away when the era of fault-tolerant quantum computation begins. So far we are in the so-called NISQ era, in which one very much has to design quantum algorithms at the machine code level and tailor the method to the exact device used. The path towards demonstrating an advantage over conventional computation is not so clear in this era, and is very much an active area of research.

The first example of a concrete and useful quantum advantage will likely occur for problems whose structure closely resembles the physics of the quantum hardware. The easiest task that one might give a quantum computer is to simulate a quantum computer (though one might dispute whether this would really constitute a simulation). The next obvious, and more useful, task might then be to simulate other quantum systems of interest. Indeed, quantum chemistry is a major focus for NISQ era applications. But can we move beyond ideas of using quantum to simulate quantum, and instead find other simulations that would suit NISQ era quantum hardware? In this paper we aim to answer this question in the affirmative, for the case of procedural generation and AI for games.

Before continuing, it is worth making a note on the scope of this work. The development of games using quantum computers can be viewed in parallel to the history of games

for conventional computers. This history began in the 1960s, with games such as *Bertie the Brain*, *Nimrod* and *OXO*. Each was a computer-based implementation of an existing game, and designed more as a showcase of technology, or for education or research, than to be fun. The 1960s brought the first unique play experiences, starting with 1962's *Spacewar!*, which allowed a player to take control of a space ship via a simulation of orbital mechanics. But it wasn't until a decade later that the first major commercial success emerged: 1972's *Pong*.

For games that used quantum computers, the last few years have reproduced some of the landmarks of the 1960s. Simple games, such as a variant of Battleships, have been made to serve as a starting point for those wanting to learn quantum programming. A game was also developed to provide an interactive way to understand benchmarking of the hardware. Many games have also been made that use quantum programming principles, but run on a simulator, in order to implement quantum-inspired. Again, this is typically done for the purposes of education. The aim of this paper is to move quantum games into the 1960s, by beginning to explore how quantum computers can actually be useful and provide unique new opportunities in game design.

II. QUANTUM NETWORKS

The field of quantum computing is not yet well-known in the study of games or of procedural generation. As such, it falls upon this paper to give a brief introduction. The aim here is not to give a comprehensive understanding of quantum computers and how they work. Instead the aim is to highlight basic properties, especially those pertinent to the approach that we will take.

A. The Qubit

Conventional digital computing is based on the notion of the 'bit': a unit of information that can take one of two possible states. Typically we refer to these states as 0 and 1. In classical information theory, there is no distinction between the internal state of the bit and its state at output. The event of extracting an output is not a defining moment in the life of a qubit. A 0 is simply a 0, and a 1 is simply a 1, whether we look at them or not.

Qubits are similarly a basic unit of information that can take one of two possible states. However, they are described by quantum information theory. This means that they are described in the same way as two-level quantum systems, such

as the spin of an electron (or a cat in a box). As such, the moment of output is a defining moment, and is represented in quantum programs by the so-called ‘measurement’ gate. It is at measurement that the qubit must decide whether it is a 0 or a 1. Before that, the output is undefined.

This behaviour is often misinterpreted to mean that quantum systems are simply random. However, though randomness can result from quantum systems, it is easier to understand them by focussing on when they give outputs with certainty.

For a bit, there is only a single way that we can extract an output: simply look at whether it is 0 or 1. For a qubit there are an infinite number of different possible methods, with the result depending on which is chosen. A set of three of these different measurements are sufficient to fully understand a qubit. They are known as x, y and z measurements.

Qubits are typically initialized in a state for which the z measurement results in the output 0 with certainty. There is similarly a state for which a z measurement will always yield the outcome 1. For x and y measurements, both of these states will lead to a completely random result. However, there are also states for which the results of x measurements are certain, with z and y being random, and similarly for y.

This is most easily described by three parameters

$$-1 \leq \langle X \rangle, \langle Y \rangle, \langle Z \rangle \leq 1.$$

These are related to the probabilities of the outcomes for x, y and z measurements, respectively. $\langle X \rangle = 1$ implies that an x measurement will return 0 with certainty, and $\langle X \rangle = -1$ implies the same for an output of 1. For $\langle X \rangle = 0$, the outcome of an x measurement is random. Corresponding behaviour is seen for $\langle Y \rangle$ and $\langle Z \rangle$ and y and z measurements.

Valid qubit states must obey the constraint

$$\langle X \rangle^2 + \langle Y \rangle^2 + \langle Z \rangle^2 \leq 1. \quad (1)$$

This immediately implies the behaviour described above. Any state that is certain of its output for one type of measurement (e.g. $\langle Z \rangle = \pm 1$ and therefore $\langle Z \rangle^2 = 1$) implies randomness for the other two ($\langle X \rangle = \langle Y \rangle = 0$). This constraint is a form of Heisenberg’s uncertainty principle.

Note that this constraint can be interpreted as requiring that qubit states must reside on or within a sphere of radius 1 centred at the origin, with $\langle X \rangle$, $\langle Y \rangle$ and $\langle Z \rangle$ serving as cartesian coordinates. This leads to a popular visualization of qubit states, known as the Bloch sphere.

The Bloch sphere is also useful for understanding the gates than can be used to manipulate a single qubit. Each can be represented as a rotation around this sphere, around a given axis by a given angle. As such, note that the value of $\langle X \rangle^2 + \langle Y \rangle^2 + \langle Z \rangle^2$ is invariant under single qubit gates. Changing this value (i.e. the distance from the origin) can only be done with the help of two-qubit or other multi-qubit gates.

B. Two Qubits

To describe a two qubit state, we need more than just the $\langle X \rangle$, $\langle Y \rangle$ and $\langle Z \rangle$ values of each qubit. We require additional

variables which will keep track of correlations between the outputs of the two qubits. For example, $-1 \leq \langle X_j X_k \rangle \leq 1$, describes the correlations that would arise between the outputs if an x measurement were made of both qubits: $\langle X_j X_k \rangle = 1$ would imply certainty of agreement, $\langle X_j X_k \rangle = -1$ would imply certainty of disagreement, and $\langle X_j X_k \rangle = 0$ would mean that any agreement or disagreements are random. There are similarly $\langle Y_j Y_k \rangle$ and $\langle Z_j Z_k \rangle$ to describe correlations between y and z measurement outcomes, $\langle X_j Y_k \rangle$ for describing correlations for an x measurement of one qubit and a y for the other, and so on.

With these new variables we obtain further constraints on valid states, for example,

$$\langle X_j P_k \rangle^2 + \langle Y_j P_k \rangle^2 + \langle Z_j P_k \rangle^2 \leq 1, P \in \{X, Y, Z\}. \quad (2)$$

In relations, these correlations can similarly be visualized as a spheres for which single qubit gates act as rotations.

Given only single qubit operations, only limited correlations can be obtained. For example, two qubits are typically initialized in the state with $\langle Z \rangle = 1$, which implies $\langle ZZ \rangle = 1$, but all other correlations are zero. Single qubit operations can change which measurement types experience this correlation, but it remains basically the same.

Sophisticated manipulation of correlations is done via two-qubit operations. We will not describe these in detail here. Instead, as an example of their effect, note that it is possible to create states such as $\langle ZZ \rangle = \langle XX \rangle = -\langle YY \rangle = 1$. Here perfect correlation (or anticorrelation) is obtained for outputs from each type of measurement, when both qubits are measured in the same way. This comes at the cost of the having $\langle X \rangle = \langle Y \rangle = \langle Z \rangle = 0$ for each qubit, which means that each qubit appears completely random when considered on its own.

The state described above is an example of what is known as an so-called entangled state. In fact, it is an example of a maximally entangled state. Entanglement exhibits a property known as monogamy, which means that maximal entanglement between two qubits prevents either from being in any way correlated with any other qubit. More generally, the degree of entanglement between any two qubits limits the degree to which either can entangle with others.

This all serves as an example of how qubits can be characterized as objects with a limited amount of certainty. They are initialized in a state that for which this certainty is directed towards the outcomes of z measurements. Single qubit gates can move this certainty to x or y measurement outcomes, or some compromise thereof. Two-qubit operations can move the certainty to collective effects of the qubits, such as correlations between their outcomes. It is the manipulation of this certainty that is the hallmark of quantum computing, rather than the remaining uncertainty.

C. Many Qubits

We have no need in this paper to introduce the general form for understanding multi-qubit states. Suffice it to say that

variables are required to keep track of all possible multi-partite correlations, for any possible subset of qubits and combination of measurements. Specifically, for n qubits the number of these variables scales as 4^n . The number of constraints on these variables also scales exponentially, but not enough to restrain the exponential number of parameters needed to fully describe an n -qubit system. Indeed, this is the reason why the simulation of quantum software becomes intractable, and hence quantum hardware will be needed.

One aspect of multi-qubit systems that will be very important for this work is the availability of two-qubit gates. This depends on the exact form of hardware used to realize the qubits. For the superconducting qubit devices that are currently most prevalent, qubits are located on a two-dimensional chip. Two-qubit gates are only possible between neighbouring qubits on this chip, and even then only for certain neighbouring pairs. This constitutes the so-called ‘coupling map’ of the device, which can differ strongly between differently designed chips.

Any coupling map corresponding to a connected graph can simulate any other without a high cost in terms of computational complexity. However, one must account for the imperfections present in near-term devices and the errors they can cause. The additional effort required to simulate alternate coupling maps will lead to the introduction of too many imperfections, causing the output to be dominated by errors. In the near-term, we should therefore stick with the coupling map we are given as much as possible in order to ensure useful results.

D. Graph-based procedural generation

In summary, near-term quantum computers can be well-described as a set of quantum objects (the qubits), placed within a network (as specified by the coupling map), interacting with their neighbours (by means of gates). It is within this structure that we must seek to encode our problems if we wish to solve them on a quantum computer. In the long-term, the encoding can be high-level and abstract. In the near-term, however, imperfections in the devices mean that more direct the encoding, the better the answer that we can obtain before the effects of errors become dominant.

If our problem is to simulate the interactions of quantum objects within a network, the encoding is clearly extremely direct. This is therefore the problem we will consider, and seek to find applications for in procedural generation.

Specifically, we will look towards the use of graphs in procedural generation. For example, for narrative generation one can use a graph to encode properties of a set of characters (the nodes) and their relationships (the edges). By looking at the current state of the graph we can see what should happen next: if two characters are found to hate each other, for example, perhaps a fight will result. After this event has taken place, the graph is updated based on the consequences. The next event can then be determined based on the new state of the graph. The evolution of the graph state therefore provides a resource for generating dynamic narratives.

When altering the state of such a graph, one must be careful to obey any consistency relations between different characteristics. For example, suppose we recorded both how friendly and how violent a character was. These are independent enough that they need to be recorded by separate variables, but they are not completely independent: it would seem contradictory to have a character that was both completely violent and completely friendly. As such, the evolution of the variables must be done in such a way that consistency is maintained.

There are many ways that this can be done. One way might be to use a network of qubits. The $\langle X \rangle$, $\langle Y \rangle$ and $\langle Z \rangle$ values of each qubit could be used to encode three characteristics of a character, which are defined such that the condition of Eq. (1) perfectly sums up their consistency relations, and single qubit quantum gates perfectly describe how their characteristics might evolve. Similarly the relationships should be defined such that they are perfectly captured by variables such as $\langle XX \rangle$ and their consistency relations, and two qubit gates perfectly describe their evolutions.

The suggestion of the last paragraph is, admittedly, quite speculative. It is not expected that the reader’s mind is now abuzz with all the ways that quantum networks would be perfectly suited to their own projects. Instead, in the rest of the paper we will explore a concrete example, to serve as a starting point for exploring how quantum networks could be used in procedural generation.

III. A THEORY OF QUANTUM GOVERNMENT

We will consider an example in which n qubits are used to simulate the actions and interactions of n entities, which we will think of as the governments of nations. In this way, the quantum network serves as a rudimentary AI. The nations are modelled very simply as having three possible kinds of action: offensive, defensive and explorative. The former two are directed at a particular neighbouring nation. In the next section we will embed these nations into a context in which these actions and their consequences are more concretely defined. However, here we consider them at an abstract level and focus instead on the implementation of the AI.

For each qubit, the $\langle X \rangle$ variable is used to encode the tendency to defend. Though the full range of this variable is $-1 \leq \langle X \rangle \leq 1$, only $0 \leq \langle X \rangle \leq 1$ will be used. Similarly $\langle Z \rangle$ encodes the tendency to aggression, and $\langle Y \rangle$ is that to explore.

Each nation will have limited capability to enact its policies. It can fully dedicate itself to one, or have some compromised sharing of priorities. In this sense it does not make sense for two of these variables to take the maximum value at once, and it makes even less sense for all three to do so. This limit will be naturally implemented via Eq. (1).

The two qubits variables describe correlations of policy between two particular nations. For example, while the $\langle X \rangle$ of each describes a generic tendency to defend, $\langle XX \rangle$ describes their specific tendency to defend against each other, and so on. The limit of the capability to enact policy against any

particular neighbour will be naturally implemented via Eq. (2). Similarly, the limited degree to which a nation can focus policy on any particular neighbour will be governed by the monogamy of entanglement.

For each nation, j , neighbour k and tactic $P \in \{X, Y, Z\}$ we calculate the following quantities.

$$E_{j,k}(P) = \langle P_j \rangle + \sum_{Q_k \in \{X, Y, Z\}} \langle P_j Q_k \rangle. \quad (3)$$

In this case, the neighbours are defined by the geopolitics of the nation's rather than the coupling map of the qubits.

With these quantities for a given nation, we then determine which is maximum. This is then the action that we can expect the nation to employ. Note that this calculation is done as if exploration is done with respect to a certain neighbour, when in fact it is not.

Calculating these quantities requires us to have access to the necessary single- and two- qubit quantities. However, as mentioned earlier, the output of qubits is in the form of bits. The quantities are therefore calculated via a tomography process, in which samples are obtained from many runs, which differ only in the form of measurement made at the end. Tomography that covers all possible pairs of qubits can be done with low overhead. We will therefore do this, to ensure that we can always access the quantities of Eq. (3) between nations that share a land border.

Once the nations have taken action against each other, the consequences should result in a change to their states. This can be done by means of single qubit operations. However, clearly the interpretation of these operations as rotations is not intuitive in this case: a nation does not typically respond to a crisis by performing a certain rotation around their aggression axis. Instead it is more natural to think of policy as evolving from its current state to (or towards) another.

This is the way that we will implement single qubit operations on our nations. Tomography tells us a nation's current state, and the actions taken by and against it determine the state to which it is moving (such as the fully defensive $\langle Y \rangle = 1$ state for a nation under attack). The single qubit gate to achieve this is then determined. This gate could be applied itself, or a partial version could be applied instead to take the state a specified fraction of the way to the target.

Note that, as mentioned earlier, single qubit operations cannot vary the value of $\langle X \rangle^2 + \langle Y \rangle^2 + \langle Z \rangle^2$. More accurately, the target state will then be the closest possible state given this restriction.

To directly manipulate correlations we can use two qubit gates. Ideally we would like a method similar to that for single qubits. For example, we might want to specify that we want $\langle ZZ \rangle = 1$ for a pair of nations on the brink of war. Then we could use two qubit tomography to determine the current state, as well as a target state that satisfies the desired condition as much as possible, while otherwise being similar to the current state. The two qubit gate to achieve this could then be found and implemented.

Attempts to implement such a system is affected by the fact that tomography does not result in a perfect description of the state of the qubits, but rather one with some statistical noise. The effects of were are found to be sufficiently negligible for the single qubit case, but were much more drastic for two qubits. As such, implementation of this 'ideal' method will be left to future work.

Instead, we will focus on using two-qubit gates in a single case: an aggressive interaction between nations causes a significant loss for one and gain for the other. A sensible consequence for this could be for the loser to change priority with aggression and defence, for the winner's aggression to remain the same. This is because the loser's tactics are clearly not working, while the winner's aggression is clearly working well.

The change in priority for the loser could be done by means of the so-called Hadamard gate. This acts on a single qubit, and can be most easily explained as swapping the values of $\langle X \rangle$ and $\langle Z \rangle$ (attack and defence in this case). It also has the effect that $\langle Y \rangle \rightarrow -\langle Y \rangle$, leading to negative values for exploration. Since this is outside the range $0 \leq \langle Y \rangle \leq 1$ that we are using, this could be a problem in general. However, assuming that such events take place only after the age of exploration has come to and end, this should not cause too many issues.

Rather than use the Hadamard alone, we will use form of controlled-Hadamard gate. This is a two-qubit gate, in which one qubit plays the role of 'control', and the other of 'target'. Simply put, this is designed such that having the control in the $\langle Z \rangle = 1$ state (certain to output 0 if a z measurement were made) a Hadamard is applied to the target. If however the control is in the $\langle Z \rangle = -1$ state (certain to output 1) nothing happens at all. For intermediate cases, the Hadamard is applied to the target in a way that is correlated to what the output of the control would be if the z measurement were made. Given that we are using $\langle Z \rangle$ to encode aggression, the effect is that the degree to which tactics change for the nation represented by the target qubit depend on exactly how aggressive that for the control qubit is.

The above description does not fully do justice to the effects of the controlled-Hadamard, especially since it implies that the control qubit is not itself affected in any case. Effects on the control include the following swapping of values, (where c and t denote the control and target qubits, respectively).

$$\langle X_c \rangle \leftrightarrow \frac{\langle X_c X_t + X_c X_t \rangle}{\sqrt{2}}$$

The previous value for the intrinsic tendency for defence of this nation therefore becomes completely dedicated to the nation of the target qubit, and vice-versa. Similarly, the gate also induces the transformations

$$\langle X_c P_k \rangle \leftrightarrow \frac{\langle X_c X_t P_k + X_c X_t P_k \rangle}{\sqrt{2}},$$

for any $P \in \{X, Y, Z\}$ for any given third nation k . Here some of the variables used to characterize the relationship between

nations c and k become tripartite quantities characterizing some mutual relationship between c , k and t . Such quantities are not used by the AI, and this previous aspect of the relationship between c and k is effectively lost. This is the consequence in this case of the monogamy of entanglement.

IV. PROCEDURAL MAP GENERATION

Using the AI system described in the previous section, we can implement a geopolitical map and history generator which simulates the growth and interactions between nations.

We will consider as an example a simple mechanic, in which everything is driven by the placement of cities. Specifically, each city has a radius of influence r , and within this radius it exerts an influence of strength $1 + \min(1, d^{-1})$, where d is the Euclidean distance on a square grid. For $r < d \leq 2r$, the influence is simply d^{-1} . Otherwise, it is zero. The ownership of each position on the map is determined by whether any nation exerts influence and, if multiple do, which exerts the most influence. This includes positions that contain cities, allowing them to change hands.

Given this mechanic, it is possible for a nation to place cities to defend against a given neighbour, by choosing their own point of least influence along the shared border. City placement for exploration is the same, but for the border with unclaimed territory. Aggressive placement against a neighbour is done by finding the border point at which their influence is least. Additionally in this case, the influence of the nation itself must be below a certain level to prevent crowding. Specifically, it is chosen to be less than the equivalent from a single city at distance $r/2$.

The number of cities that a nation has depends on its area, A . Specifically, $A/(\pi r^2)$ is used. If a nation deems it necessary to place a further city once this limit is reached, an existing city must be razed. In this case, the city at the position of maximum influence (other than the capital) is chosen. No further city can be built on the ruins.

Information regarding gains and loss of territory is fed back into the AI. The way this is done depends on the length of the border with unexplored territory, f , the area lost, A_l , and the area gained from other nations, A_g . Whichever of these is greater for each nation determines whether its qubit is moved toward the fully explorative, defensive or aggressive state, respectively. In the latter two cases, the fraction of the rotation towards this state that is performed is the ratio of the lost or gained area with πr^2 . In the former, the fraction is always $1/4$. For the transfer of a city, the effect on the AI is performed using the two-qubit gate described in the last section. The nation that lost the city is the target, and the winner is the control.

With these rules for city placement and the use of the AI, the procedure for map generation is complete. Note that the procedural could also be used to implement a game, simply by handing control over city placement for one or more nations to a player. In this case, the quantum network serves as the AI for the non-player nations, and could provide an advisor for player controlled nations.

V. RESULTS

VI. CONCLUSIONS

ACKNOWLEDGMENT

REFERENCES