エリッツァー・ベイドマンが戦艦を一発も撃たずに勝利

注:これはmedium paperの量子戦艦ではありません!

<https://decodoku.medium.com/quantum-battleships-the-first-multiplayer-game-for-a-quantum-computer-e4d600ccb3f3>

背景と動機

1990 年代は量子コンピューティングにとって極めて重要な時代でした。1994年、ピーター・ショールは彼の名前を冠したアルゴリズムを発表し[1]、量子力学が計算上の利点をどのように提供できるかを実証しました。わずか1年後の1995年、デビッド・ワインランドのチームは、量子コンピューティングの基礎であるトラップイオン[2]を使用して最初の量子論理ゲートを実装することで記念碑的な偉業を達成しました。革新は続き、1997年にAnton Zeilingerのチームは量子テレポーテーションの最初の実験的デモンストレーションを発表しました[3]。

これらの進歩の中で、1993 年に Avshalom Elitzur と Lev Vaidman は、量子力学の基本概念であるもつれ、重ね合わせ、状態崩壊という中核的な概念を探求するために設計された示唆に富む爆弾テスター実験を提案しました [4]。彼らの劇的な爆弾テスターの思考実験では、マッハツェンダー干渉計を使用して爆弾を爆発させることなく爆弾の存在を検出しました。これは革命的な概念であり、直接の相互作用なしにオブジェクトに関する情報を取得する能力でした。

このハッカソンでは、Elitzur-Vaidman 爆弾テスターに基づいて構築することに挑戦します。あなたの使命は、この非インタラクティブなプロービング方法を戦艦ゲームグリッドに拡張し、すべての船を「ぶつける」ことなく見つけることを目指します。従来の戦艦では、船を見つけるということは、直撃または外れを意味します。ただし、Elitzur-Vaidman 爆弾テスターは、相互作用なしで物体を探査する量子力学の力を示しています。このハッカトンでは、核となる量子概念を活用して戦艦グリッド上のすべての船を見つけ、一発の発砲する前に勝利を確実にする量子回路を開発します。

はじめ

このハッカソンチャレンジは、次の3つの部分に分かれています。

1. オリジナルのElitzur-Vaidman爆弾テスターを量子回路に変換する方法を学びます。
2. オリジナルの Elitzur-Vaidman 爆弾テスターをアップグレードして、爆弾 (敵の船) と対話することなく 100% 検出に近づける方法を学びましょう。これにより、動的回路および/または繰り返しによるより優れたElitzur-Vaidman設計を利用できます[5]。
3. 本当の課題は、これらの概念を類似の戦艦ゲームに適用し、この場合、ランダムな戦艦グリッド [~16 量子ビットで動作するオペレーター] (ユーザーには非表示) を提供し、操作せずに船の位置を検出する必要があります。

出力回路は、いくつかの異なるランダムな戦艦シナリオでグレーダー機能で評価され、検出された船の数をランク付けします。

推奨されるリソース

[1] ショール、PW (1994)。量子計算のためのアルゴリズム 離散対数と因数分解議事録、コンピュータの基礎に関する第 35 回年次シンポジウム €i0science €i0(pp. 124-134)。IEEEです。

[2] モンロー、C.、ミークホフ、DM、キング、BE、板野、WM、ワインランド、DJ (1995)。基本量子論理ゲートの実証€i0フィジカルレビューレターズ、75(25)、4714-4717。

[3] Bouwmeester, D.、Pan, JW、Mattle, K.、Eibl, M.、Weinfurter, H.、および Zeilinger, A. (1997)。実験的量子テレポーテーションネイチャー、390(6660)、575-579。

[4] エリッツール、AC、およびベイドマン、L. (1993)。量子力学的相互作用のない測定物理学の基礎、23(7)、987-997。

[5] ペレイラ・ピント、V.、ペレイラ・デ・オリベイラ、B.、安岡光江、FM、クルテイユ、PW、カイアド・デ・カストロ・ネト、J. (2023)。Elitzur‐Vaidman爆弾実験問題による量子理解の探求ブラジル物理学ジャーナル、53(6)、152。

ウィキペディアからの一般的なエリッツール・ヴァイドマンドの説明: <https://en.wikipedia.org/wiki/Elitzur%E2%80%93Vaidman_bomb_tester>

動的回路の使用を実証するペニーレーン演習: <https://pennylane.ai/codebook/pennylane-fundamentals/dynamic-circuits>

複数の爆弾への拡張をブログでデモンストレーション: <https://algassert.com/post/1700>