



**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(11) 공개번호 10-2021-0129202  
(43) 공개일자 2021년10월27일

- (51) 국제특허분류(Int. Cl.)  
G06N 10/00 (2019.01) G06N 5/00 (2019.01)  
G06N 5/02 (2006.01) B82Y 10/00 (2017.01)
- (52) CPC특허분류  
G06N 10/00 (2019.01)  
G06N 5/003 (2019.01)
- (21) 출원번호 10-2021-7031497
- (22) 출원일자(국제) 2020년04월06일  
심사청구일자 2021년09월30일
- (85) 번역문제출일자 2021년09월30일
- (86) 국제출원번호 PCT/EP2020/059733
- (87) 국제공개번호 WO 2020/207956  
국제공개일자 2020년10월15일
- (30) 우선권주장  
62/831,418 2019년04월09일 미국(US)  
(뒷면에 계속)

- (71) 출원인  
인터내셔널 비즈니스 머신즈 코퍼레이션  
미국 10504 뉴욕주 아몬크 뉴오차드 로드
- (72) 발명자  
그린버그, 돈  
미국 뉴욕 10598, 요크타운 하이츠, 키차완 로드 1101, 피오 박스 218, 아이비엠 코퍼레이션  
피스토야, 마르코  
미국 뉴욕 10598, 요크타운 하이츠, 피오 박스 218, 루트 134, 키차완 로드 1101, 인터랙츄얼 프 로퍼티 로, 아이비엠 코퍼레이션  
(뒷면에 계속)
- (74) 대리인  
허정훈

전체 청구항 수 : 총 26 항

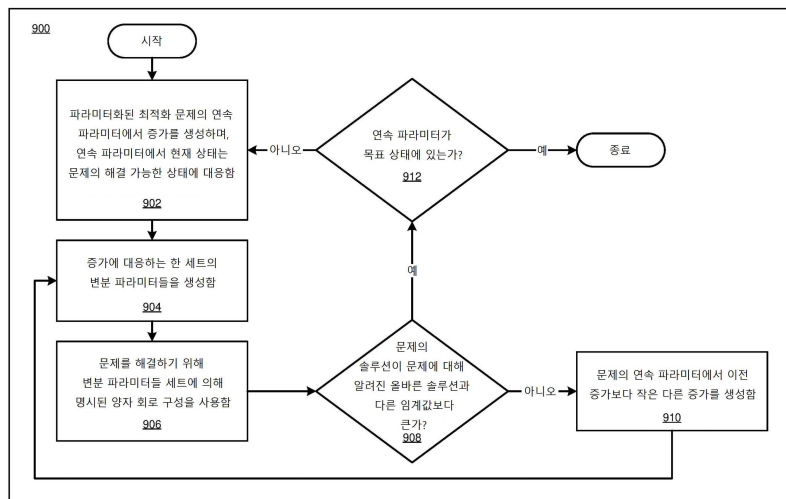
(54) 발명의 명칭 양자 컴퓨팅에서 어려운 변분 양자 문제를 해결하기 위해 중간 재-최적화를 사용하는 단열 진행

**(57) 요약**

양자 컴퓨팅의 어려운 변분 양자 문제를 해결하기 위한 중간 재-최적화에 의한 단열 진행(ADIABATIC PROGRESSION WITH INTERMEDIATE RE-OPTIMIZATION TO SOLVE HARD VARIATIONAL QUANTUM PROBLEMS IN QUANTUM COMPUTING)

변분 문제에 대응하는 양자 회로를 실행하기 위한 하이브리드 고전-양자 컴퓨팅 디바이스가 구성된다. 상기 구성하는 단계는 상기 하이브리드 고전-양자 컴퓨팅 디바이스가 단열 진행 연산을 수행함으로써 상기 양자 회로를 실행하게 하는 단계를 더 포함하고, 상기 단열 진행 연산은 상기 변분 문제의 난이도를 상기 문제의 단순화된 버전으로부터 상기 변분 문제로 증가시키는 단계를 포함한다.

**대표도** - 도9



(52) CPC특허분류

*G06N 5/02* (2019.01)

*B82Y 10/00* (2013.01)

(72) 발명자

**첸, 리차드**

미국 뉴욕 10598, 요크타운 하이츠, 키차완 로드  
1101, 피오 박스 218, 아이비엠 코포레이션

**난니치니, 지아코모**

미국 뉴욕 10598, 요크타운 하이츠, 키차완 로드  
1101, 피오 박스 218, 아이비엠 코포레이션

(30) 우선권주장

16/427,043 2019년05월30일 미국(US)

16/427,080 2019년05월30일 미국(US)

## 명세서

### 청구범위

#### 청구항 1

컴퓨터-구현 방법에 있어서, 상기 방법은:

변분 문제(variational problem)에 대응하는 양자 회로를 실행하도록 하이브리드 고전-양자 컴퓨팅 디바이스(a hybrid classical-quantum computing device)를 구성하는 단계를 포함하고: 상기 구성하는 단계는:

상기 하이브리드 고전 양자-컴퓨팅 디바이스가 단열 진행 연산(adiabatic progression operation)을 수행함으로써 상기 양자 회로를 실행하게 하는 단계;

상기 변분 문제에 대응하는 파라미터화된 최적화 문제의 연속 파라미터(a continuous parameter of a parametrized optimization problem)에서 제1 증가를 생성하는 단계-상기 연속 파라미터의 현재 상태는 상기 최적화 문제의 해결 가능한 상태(a solvable state)에 대응함-;

상기 제1 증가에 대응하는 한 세트의 변분 파라미터들을 생성하는 단계-; 상기 변분 파라미터들 세트는 상기 양자 회로의 구성을 명시함-; 및

상기 양자 회로의 출력이 상기 문제의 알려진 올바른 솔루션 (a known correct solution of the problem)과 임계량보다 적거나 또는 동일한 양만큼(by less than or equal to a threshold amount) 다르다고 결정하는 것에 응답하여, 상기 연속 파라미터의 현재 상태를 상기 제1 증가로 대체하는 단계를 포함하는

컴퓨터-구현 방법.

#### 청구항 2

제1항에 있어서, 상기 하이브리드 클래식-양자 컴퓨팅 디바이스는 노이즈 중간 규모 양자(Noisy Intermediate Scale Quantum: NISQ) 디바이스를 포함하는

컴퓨터-구현 방법.

#### 청구항 3

제2항에 있어서 상기 방법은:

상기 제1 증가에 대응하는 상기 변분 파라미터들 세트를 사용하여, 정사각형 양자 회로(a square quantum circuit)를 생성하는 단계-상기 정사각형 양자 회로는 상기 NISQ 디바이스 상에서 구현하기에 적합함-를 더 포함하는

컴퓨터-구현 방법.

#### 청구항 4

제1항에 있어서, 상기 방법은:

상기 변분 문제의 난이도(the difficulty)를 진행하는 속도(pace)를 컨트롤하는 단계를 더 포함하는

컴퓨터-구현 방법.

#### 청구항 5

제4항에 있어서, 상기 제1 증가는 상기 연속 파라미터의 목표 상태(a goal state)와 상기 연속 파라미터의 현재 상태 사이의 차이를 포함하는

컴퓨터-구현 방법.

#### 청구항 6

제4항에 있어서, 상기 제1 증가는 상기 연속 파라미터의 목표 상태와 상기 연속 파라미터의 현재 상태 사이의 차이(a difference)를 포함하고, 그 다음 상기 차이는, 양의 정수(a positive whole number)인  $n$ 에 의해서 나누어지는

컴퓨터-구현 방법.

#### 청구항 7

제4항에 있어서, 상기 방법은:

상기 양자 회로의 출력이 상기 최적화 문제의 알려진 올바른 솔루션 (a known correct solution of the optimization problem)과 임계량보다 많은 양만큼(by more than a threshold amount) 다르다고 결정하는 것에 응답하여, 상기 현재 상태에서부터 상기 연속 파라미터에서 제2 증가를 생성하는 단계 - 상기 제2 증가는 상기 제1 증가보다 작음-를 더 포함하는

컴퓨터-구현 방법.

#### 청구항 8

제7항에 있어서, 상기 제2 증가는 상기 제1 증가의  $1/n$ ( $1/n$  multiplied by the first increase)-  $n$ 은 양의 정수임-인

컴퓨터-구현 방법.

#### 청구항 9

컴퓨터 사용 가능 프로그램 제품에 있어서, 상기 컴퓨터 사용 가능 프로그램 제품은 하나 또는 그 이상의 컴퓨터-판독가능 스토리지 디바이스들, 및 상기 하나 또는 그 이상의 스토리지 디바이스 중 적어도 하나에 저장된 프로그램 명령들 포함하고, 상기 저장된 프로그램 명령들은:

변분 문제(variational problem)에 대응하는 양자 회로를 실행하도록 하이브리드 고전-양자 컴퓨팅 디바이스(a hybrid classical-quantum computing device)를 구성하는 단계를 수행하는 프로그램 명령들을 포함하고: 상기 구성하는 단계는:

상기 하이브리드 고전 양자-컴퓨팅 디바이스가 단일 진행 연산(adiabatic progression operation)을 수행함으로써 상기 양자 회로를 실행하게 하는 단계;

상기 변분 문제에 대응하는 파라미터화된 최적화 문제의 연속 파라미터(a continuous parameter of a parametrized optimization problem)에서 제1 증가를 생성하는 단계-상기 연속 파라미터의 현재 상태는 상기 최적화 문제의 해결 가능한 상태(a solvable state)에 대응함-;

상기 제1 증가에 대응하는 한 세트의 변분 파라미터들을 생성하는 단계-; 상기 변분 파라미터들 세트는 양자 회로의 구성을 명시함-; 및

상기 양자 회로의 출력이 상기 문제의 알려진 올바른 솔루션(a known correct solution of the problem)과 임계량보다 적거나 또는 동일한 양만큼(by less than or equal to a threshold amount) 다르다고 결정하는 것에 응답하여, 상기 연속 파라미터의 현재 상태를 상기 제1 증가로 대체하는 단계를 포함하는

컴퓨터 사용 가능 프로그램 제품.

#### 청구항 10

제9항에 있어서, 상기 하이브리드 클래식-양자 컴퓨팅 디바이스는 노이즈 중간 규모 양자(Noisy Intermediate Scale Quantum: NISQ) 디바이스를 포함하는

컴퓨터 사용 가능 프로그램 제품.

#### 청구항 11

제10항에 있어서, 상기 프로그램 명령들은:

상기 제1 증가에 대응하는 상기 변분 파라미터들 세트를 사용하여, 정사각형 양자 회로(a square quantum

circuit)를 생성하는 단계를 수행하는 프로그램 명령들-상기 정사각형 양자 회로는 상기 NISQ 디바이스 상에서 구현하기에 적합함-을 더 포함하는

컴퓨터 사용 가능 프로그램 제품.

#### 청구항 12

제9항에 있어서, 상기 프로그램 명령들은:

상기 변분 문제의 난이도(the difficulty)를 진행하는 속도를 컨트롤하는 단계를 수행하는 프로그램 명령들을 더 포함하는

컴퓨터 사용 가능 프로그램 제품.

#### 청구항 13

제12항에 있어서, 상기 제1 증가는 상기 연속 파라미터의 목표 상태(a goal state)와 상기 연속 파라미터의 현재 상태 사이의 차이를 포함하는

컴퓨터 사용 가능 프로그램 제품.

#### 청구항 14

제12항에 있어서, 상기 제1 증가는 상기 연속 파라미터의 목표 상태와 상기 연속 파라미터의 현재 상태 사이의 차이(a difference)를 포함하고, 그 다음 상기 차이는, 양의 정수(a positive whole number)인  $n$ 에 의해서 나누어지는

컴퓨터 사용 가능 프로그램 제품.

#### 청구항 15

제12항에 있어서, 상기 프로그램 명령들은:

상기 양자 회로의 출력이 상기 최적화 문제의 알려진 올바른 솔루션 (a known correct solution of the optimization problem)과 임계량보다 많은 양만큼(by more than a threshold amount) 다르다고 결정하는 것에 응답하여, 상기 현재 상태에서부터 상기 연속 파라미터에서 제2 증가를 생성하는 단계를 수행하는 프로그램 명령들 - 상기 제2 증가는 상기 제1 증가보다 작음-을 더 포함하는

컴퓨터 사용 가능 프로그램 제품.

#### 청구항 16

제15항에 있어서, 상기 제2 증가는 상기 제1 증가의  $1/n$ ( $1/n$  multiplied by the first increase)-  $n$ 은 양의 정수임-인

컴퓨터 사용 가능 프로그램 제품.

#### 청구항 17

제9항에 있어서, 상기 컴퓨터 사용 가능 코드는 데이터 처리 시스템의 컴퓨터 관독 가능 스토리지 디바이스에 저장되고, 상기 컴퓨터 사용 가능 코드는 원격 데이터 처리 시스템으로부터 네트워크를 통해 전송되는,

컴퓨터 사용 가능 프로그램 제품.

#### 청구항 18

제9항에 있어서, 상기 컴퓨터 사용 가능 코드는 서버 데이터 처리 시스템의 컴퓨터 관독 가능 스토리지 디바이스에 저장되고, 상기 컴퓨터 사용 가능 코드는 원격 데이터 처리 시스템과 연관된 컴퓨터 관독 가능 스토리지 디바이스에서 사용하기 위해 상기 원격 데이터 처리 시스템으로 네트워크를 통해 다운로드 되며, 상기 프로그램 명령들은:

상기 요청과 연관된 컴퓨터 사용 가능 코드의 사용을 계량(meter)하는 단계를 수행하는 프로그램 명령들; 및

상기 계량된 사용에 기초하여 청구서를 생성하는 단계를 수행하는 프로그램 명령들을 더 포함하는 컴퓨터 사용 가능 프로그램 제품.

**청구항 19**

컴퓨터 시스템에 있어서, 상기 컴퓨터 시스템은 하나 혹은 그 이상의 프로세서들, 하나 혹은 그 이상의 컴퓨터 관독-가능 메모리들, 하나 혹은 그 이상의 컴퓨터 관독-가능 스토리지 디바이스들, 및 상기 하나 혹은 그 이상의 컴퓨터 관독-가능 메모리들 중 적어도 하나를 통해 상기 하나 또는 그 이상의 프로세서들 중 적어도 하나에 의한 실행을 위한 상기 하나 또는 그 이상의 스토리지 디바이스들 중 적어도 하나에 저장된 프로그램 명령들을 포함하고, 상기 저장된 프로그램 명령들은:

변분 문제(variational problem)에 대응하는 양자 회로를 실행하도록 하이브리드 고전-양자 컴퓨팅 디바이스(a hybrid classical-quantum computing device)를 구성하는 단계를 수행하는 프로그램 명령들을 포함하고: 상기 구성하는 단계는:

상기 하이브리드 고전 양자-컴퓨팅 디바이스가 단일 진행 연산(adiabatic progression operation)을 수행함으로써 상기 양자 회로를 실행하게 하는 단계;

상기 변분 문제에 대응하는 파라미터화된 최적화 문제의 연속 파라미터(a continuous parameter of a parametrized optimization problem)에서 제1 증가를 생성하는 단계- 상기 연속 파라미터의 현재 상태는 상기 최적화 문제의 해결 가능한 상태(a solvable state)에 대응함 -;

상기 제1 증가에 대응하는 한 세트의 변분 파라미터들을 생성하는 단계-; 상기 변분 파라미터들 세트는 양자 회로의 구성을 명시함-; 및

상기 양자 회로의 출력이 상기 문제의 알려진 올바른 솔루션 (a known correct solution of the problem)과 임계량보다 적거나 또는 동일한 양만큼(by less than or equal to a threshold amount) 다르다고 결정하는 것에 응답하여, 상기 연속 파라미터의 현재 상태를 상기 제1 증가로 대체하는 단계를 포함하는

컴퓨터 시스템.

**청구항 20**

제19항에 있어서, 상기 하이브리드 클래식-양자 컴퓨팅 디바이스는 노이즈 중간 규모 양자(Noisy Intermediate Scale Quantum: NISQ) 디바이스를 포함하는 상기

컴퓨터 시스템.

**청구항 21**

제20항에 있어서, 상기 컴퓨터시스템은:

상기 제1 증가에 대응하는 상기 변분 파라미터들 세트를 사용하여, 정사각형 양자 회로(a square quantum circuit)를 생성하는 단계를 수행하는 프로그램 명령들- 상기 정사각형 양자 회로는 상기 NISQ 디바이스 상에서 구현하기에 적합함 -을 더 포함하는

컴퓨터 시스템.

**청구항 22**

제19항에 있어서, 상기 컴퓨터 시스템은:

상기 변분 문제의 난이도(the difficulty)를 진행하는 속도를 컨트롤하는 단계를 수행하는 프로그램 명령들을 더 포함하는

컴퓨터 시스템.

**청구항 23**

제22항에 있어서, 상기 제1 증가는 상기 연속 파라미터의 목표 상태(a goal state)와 상기 연속 파라미터의 현재 상태 사이의 차이를 포함하는

컴퓨터 시스템.

**청구항 24**

제22항에 있어서, 상기 제1 증가는 상기 연속 파라미터의 목표 상태와 상기 연속 파라미터의 현재 상태 사이의 차이(a difference)를 포함하고, 그 다음 상기 차이는, 양의 정수(a positive whole number)인 n에 의해서 나누어지는

컴퓨터 시스템.

**청구항 25**

제22항에 있어서, 상기 컴퓨터 시스템은:

상기 양자 회로의 출력이 상기 최적화 문제의 알려진 올바른 솔루션 (a known correct solution of the optimization problem)과 임계량보다 많은 양만큼(by more than a threshold amount) 다르다고 결정하는 것에 응답하여, 상기 현재 상태에서부터 상기 연속 파라미터에서 제2 증가를 생성하는 단계를 수행하는 프로그램 명령들 - 상기 제2 증가는 상기 제1 증가보다 작음-을 더 포함하는

컴퓨터 시스템.

**청구항 26**

컴퓨터 프로그램에 있어서, 상기 컴퓨터 프로그램은 상기 프로그램이 컴퓨터 상에서 실행될 때 제1항 내지 제8항의 항들 중 어느 한 항의 방법을 수행하도록 구성된 프로그램 코드 수단을 포함하는

컴퓨터 프로그램.

**발명의 설명**

**기술 분야**

[0001] 이 출원은, 전체로서 참조로 포함되는, "양자 컴퓨팅의 어려운 변분 양자 문제를 해결하기 위한 중간 재-최적화를 사용하는 단일 진행(Adiabatic Progression with Intermediate Re-Optimization to Solve Hard Variational Quantum Problems in Quantum Computing)"이라는 제목으로 2019년 4월 9일자로 출원된 미국 가출원 62831418호에 대해 우선권을 주장한다. 본 출원은 또한 미국 출원 번호 16/427043 및 미국 출원 번호 16/427080에 대해 우선권을 주장한다.

**배경 기술**

[0002] 본 발명은 일반적으로 특정 유형들의 문제들을 해결하기 위해 양자 컴퓨팅 데이터 처리 환경을 운영하는 방법, 시스템, 및 컴퓨터 프로그램 제품에 관한 것이다. 보다 상세하게는, 본 발명은 양자 컴퓨팅에서 어려운 변분 양자 문제들(hard variational quantum problems)을 해결하기 위해 중간 재-최적화를 사용하는 단일 진행을 위한(for adiabatic progression with intermediate re-optimization) 방법, 시스템, 및 컴퓨터 프로그램 제품에 관한 것이다.

[0003] 이하, 명세서의 문구의 단어에서 "Q" 접두어는 명백하게 구별되게 표시하지 않는 한 양자 컴퓨팅 컨텍스트에서 단어 또는 문구를 참조함을 표시한다.

[0004] 분자들과 아원자 입자들(Molecules and subatomic particles)은 양자역학의 법칙을 따르며, 양자역학은 물리 세계가 가장 근본적인 수준들에서 어떻게 작동하는지를 탐구하는 물리학의 한 분야이다. 이 레벨에서 입자들은 둘 이상의 상태를 동시에 띠며 매우 멀리 있는 다른 입자와 상호작용하면서 이상한 방식으로 동작한다. 양자 컴퓨팅은 이러한 양자 현상을 이용하여 정보를 처리한다.

[0005] 오늘날 사용하는 컴퓨터들을 고전적인 컴퓨터들(여기서는 "종래의" 컴퓨터들 또는 종래의 노드들. 또는 "CN"이라고도 함)라고 한다. 종래의 컴퓨터는 반도체 재료들과 기술을 사용하여 제작된, Von Neumann 아키텍처로 알려져 있는, 종래의 프로세서, 반도체 메모리, 자기 또는 솔리드-스테이트 스토리지 디바이스를 사용한다. 특히 종래 컴퓨터들에서 프로세서들은 이진 프로세서들로서, 즉, 1과 0으로 표시된 이진 데이터에 대

해 연산한다.

- [0006] [0006] 양자 프로세서(q-processor)는 컴퓨터 작업들을 수행하기 위해 얽힌 큐비트 디바이스들(여기서는 간단히 "큐비트", 복수 "큐비트들" 모두를 가리킴)의 특이한 특성(the odd nature of entangled qubit devices)을 사용한다. 양자역학이 작동하는 특정한 영역에서, 물질의 입자들은, "온(on)" 상태, "오프(off)" 상태, 그리고 동시에 "온" 상태 및 "오프" 상태와 같은, 여러 상태들로 존재할 수 있다. 반도체 프로세서를 이용한 이진 컴퓨팅은 단지 온과 오프 상태(이진 코드의 경우 1과 0에 해당)만 사용하도록 제한되지만, 양자 프로세서는 물질의 이들 양자 상태들을 이용하여 데이터 컴퓨팅에 사용할 수 있는 신호를 출력한다.
- [0007] [0007] 종래 컴퓨터는 정보를 비트들로 인코딩 한다. 각 비트는 1 또는 0의 값을 취할 수 있다. 이들 1들과 0들은 궁극적으로 컴퓨터 기능을 구동하는 온/오프 스위치 역할을 한다. 반면에 양자 컴퓨터는 양자 물리학의 두 가지 핵심 원리인 중첩과 얽힘(superposition and entanglement)에 따라 작동하는 큐비트를 기반으로 한다. 중첩은 각 큐비트가 1과 0을 동시에 나타낼 수 있음을 의미한다. 얽힘은 하나의 중첩에 있는 큐비트들이 비-고전적 방식으로(in a non-classical way) 서로 상관될 수 있음을 의미하며, 즉, 하나의 상태(1 또는 0 또는 둘 모두)는 다른 하나의 상태에 따라 달라질 수 있고, 두 개의 큐비트들이 개별적으로 처리될 때보다 서로 얽혔을 때 두 개의 큐비트들에 관해 확인될(ascertained) 수 있는 정보가 더 많다.
- [0008] [0008] 이들 두 개의 원리들을 사용하여, 큐비트들은 보다 정교한 정보의 프로세서들로서 작동하며, 이는 양자 컴퓨터들이 종래의 컴퓨터를 사용하여 다루기 곤란한 어려운 문제들을 해결할 수 있도록 해주는 방식으로 작동할 수 있게 한다. IBM은 초전도 큐비트를 사용하여 양자 프로세서의 작동 가능성을 성공적으로 구축하고 보여주었다(IBM은 미국 및 기타 국가에서 인터넷서널 비즈니스 머신즈 코퍼레이션의 등록 상표이다).
- [0009] [0009] 초전도 큐비트는 조셉슨 접합부(a Josephson junction)를 포함한다. 조셉슨 접합부는 비-초전도 물질에 의해서 두 개의 박막 초전도 금속 층들을 분리함으로써 형성된다. 초전도 층들의 금속이 초전도 상태가 되면(예를 들어, 금속의 온도를 특정 극저온(cryogenic temperature)으로 낮춤에 의해서) 전자 쌍들이 한 초전도 층으로부터 비-초전도 층을 통해 다른 초전도 층으로 터널링 할 수 있다. 하나의 큐비트에서, 분산형 비선형 인덕터(a dispersive nonlinear inductor)로서 기능을 하는 조셉슨 접합부는 비선형 마이크로파 오실레이터를 형성하는 하나 또는 그 이상의 정전 디바이스들(capacitive devices)과 전기적으로 병렬로 결합된다. 상기 오실레이터는 큐비트 회로의 인덕턴스 및 캐패시턴스 값에 의해서 결정되는 공진/전이 주파수를 갖는다. "큐비트"라는 용어는 사용되는 곳에서 명확하게 구별하여 사용되지 않는 한 조셉슨 접합부를 채용하는 초전도 큐비트 회로(a superconducting qubit circuitry)를 가리킨다.
- [0010] [0010] 상기 큐비트들에 의해 처리된 정보는 마이크로파 주파수들 범위에서 마이크로파 신호들/광자들의 형태로 전달되거나 전송된다. 상기 마이크로파 신호들은 캡처되고, 처리되며, 분석되어 거기에 인코딩 된 양자 정보가 해독된다. 판독 회로(A readout circuit)는 상기 큐비트와 결합된 회로이고, 큐비트의 양자 상태를 캡처하고, 읽고 그리고 측정한다. 상기 판독 회로의 출력은 q-프로세서가 계산들을 수행하는 데 사용할 수 있는 정보이다.
- [0011] [0011] 초전도 큐비트는 두 개의 양자 상태들  $|0\rangle$  및  $|1\rangle$ 을 갖는다. 이 두 개의 상태들은 원자들의 두 개의 에너지 상태들, 예를 들어, 초전도 인공 원자의 접지( $|g\rangle$ ) 및 제1 여기 상태( $|e\rangle$ )(the ground ( $|g\rangle$ ) and first excited state ( $|e\rangle$ ))(초전도 큐비트) 일 수 있다. 다른 예들은 핵 또는 전자 스핀들의 스핀-업 및 스핀-다운 (spin-up and spin-down of the nuclear or electronic spins), 결정 결합의 두 위치들 및 양자 도트의 두 상태들(two states of a quantum dot)를 포함한다. 상기 시스템은 양자 성질을 갖고 있기 때문에 두 개의 상태들의 모든 조합이 허용되고 유효하다.
- [0012] [0012] 큐비트들을 사용하는 양자 컴퓨팅을 신뢰할 수 있으려면, 예를 들어 큐비트들 자체, 큐비트들 관련 판독 회로 및 양자 프로세서의 다른 부분들이 큐비트의 에너지 상태들을, 에너지를 주입하거나 또는 발산시키는 것과 같은 심각한 방식으로 변경시키거나 또는 큐비트의  $|0\rangle$  및  $|1\rangle$  사이의 상대 위상에 영향을 주어서는 안된다. 양자 정보로 동작하는 모든 회로에 대한 이러한 동작상의 제약(operational constraint)은 그러한 회로에 사용되는 반도체 및 초전도 구조물들의 조립에 특별한 고려를 필요로 한다.
- [0013] [0013] 양자 컴퓨팅은 종종 종래의 컴퓨팅에서 보다 더 빠르게 문제들을 해결하기 위해 사용될 수 있다. 예를 들어, 양자 컴퓨팅에 사용되는 한 가지 양자 알고리즘은 그로버의 검색(Grover's Search)이며, 이는 종래 컴퓨팅의 경우보다 더 적은 수의 검색들을 사용하여 N 항목들의 정렬되지 않은 목록을 검색한다.
- [0014] [0014] 양자 게이트들은 양자 계산을 위한 기본 빌딩 블록들이며, 이들은 고전적 논리 게이트들이 비트들에 관해 작용하는 방식으로 큐비트들에 작용하는데, 한 번에 하나 및 둘에 작용하여, 큐비트의 상태들을 컨트롤 가능

한 방식으로 변경한다. X 게이트는, 고전 컴퓨팅에서 NOT 게이트가 단일 비트의 상태를 반전시키는 것과 마찬가지로, 단일 큐비트의 상태를 반전시킨다. H 게이트, 또는 하다마드 (Hadamard) 게이트는 단일 큐비트를 0과 1 양자 상태들의 조합인, 중첩 상태가 되게 한다. 큐비트는 측정될 때 명확한 상태가 결정된다(resolve). 예를 들어, 0의 양자 상태를 갖는 입력이 제공되면, 하다마드 게이트 내에서 양자 상태는 중첩상태에 있게 되지만, 출력은 양자 0 상태에 있을 확률이 50 퍼센트이고 양자 1 상태에 있을 확률이 50 퍼센트이다. 다른 단일-큐비트 게이트들은 다른 정의된 방식들로 큐비트 상태를 변경한다.

[0015] [0015] 다수-큐비트 게이트들은 큐비트들 간에 조건 논리(conditional logic)를 수행하는 게이트들을 구현하는데, 이는 한 큐비트의 상태가 다른 큐비트의 상태에 따라 달라짐을 의미한다. 예를 들어, 컨트롤된 NOT(Controlled-NOT), 또는 CNOT 게이트는, 타겟 큐비트와 컨트롤 큐비트라는 두 개의 큐비트들을 갖는다. 만일 컨트롤 큐비트가 1의 양자 상태에 있으면, CNOT 게이트는 타겟 큐비트의 상태를 반전시킨다. 만일 컨트롤 큐비트가 0의 양자 상태에 있으면, CNOT 게이트는 타겟 큐비트의 상태를 변경하지 않는다.

[0016] [0016] 다수의 큐비트들은 또한 서로 얽힐 수 있다(entangled). 두 개 혹은 그 이상의 큐비트들이, 서로에게 영향을 미치기에는 너무 멀리 떨어져 있음에도 불구하고, 개별적으로는 랜덤하지만, 각 개체가 서로 독립적이라고 하기에는 너무 강하게 상관되어 있는 방식으로 작동할 때 그들은 얽혀 있다고 할 수 있다. 그 결과, 얽힌 다수-큐비트 시스템의 결합된 속성들은 예측될 수 있지만, 그러한 시스템에서 각 개별 큐비트를 측정하는 개별 결과는 예측될 수 없다.

[0017] [0017] 종래 컴퓨팅과 유사하게, 양자 컴퓨팅 게이트들은, 양자 회로들이라고 하는 더 큰 그룹들로 조립되어, 더 복잡한 작업들을 수행할 수 있다. 예를 들어, 한 쌍의 큐비트들 상태를 교환하는, SWAP 게이트는, 세 개의 CNOT 게이트들로 구축될 수 있다.

[0018] [0018] 양자 회로들은 일부 연산들을 병렬로 수행하고 일부는 직렬로 수행할 수 있다. 프로그램에서 가장 긴 시리즈의 길이를 양자 회로의 깊이라고도 한다. 예를 들어, SWAP 게이트를 구성하는 3개의 CNOT 게이트들은 직렬로 배열되고, 깊이는 3이다. 얇은 깊이의 프로그램들은 실행 시간이 덜 걸리고 더 나은 성능을 제공하므로, 선호된다.

**발명의 내용**

**해결하려는 과제**

[0019] [0019] 종래 컴퓨터들은 프로세서-특정 어셈블리 언어들로 제공되는 것과 같은 특정 명령 단계들로 수작업-프로그램 될(hand-programmed) 필요가 없다. 대신, 프로그래머들은 하드웨어-독립적인 코드를 고급 언어로 작성하고, 컴파일러는 이 코드를 특정 프로세서에서 실행할 수 있도록 어셈블리 언어로 변환한다. 유사하게, 양자 컴퓨팅에서도 프로그래머들은 개별 게이트들을 명시할 필요가 없다. 대신 프로그래머들은 고급 언어로 고급 코드를 작성할 수 있다. 컴파일러는 이 코드를 구문 분석하고(parse) 그 것을 양자 회로에 매핑한다. 최종으로, 양자 프로세서는 양자 회로를 실행한다. 양자 프로그래머들은 또한 화학, 인공 지능 및 최적화와 같은 영역들에서 문제들을 해결하는 데 사용하기 위해, 이미-프로그래밍된 라이브러리들을 사용할 수 있다.

[0020] [0020] 문제들에는 변분 문제들(variational problems)이라고 하는 클래스의 문제들이 존재하며, 이들은 어려운 변분 문제들(hard variational problems) 과 쉬운 변분 문제들(easy variational problems)의 서브클래스로 나누어 진다. 최적화 문제는 변분 문제들의 하나의 비-제한적인 예이고, 이는 차원의 수(dimensionality), 노드들의 수, 원하는 정확도 레벨, 및 기타 요인들에 따라, 어렵거나 쉬울 수 있다(hard or easy). 최적화 문제는 여러 가능한 솔루션들을 갖는 문제에서 가장 좋은 또는 최적의 솔루션을 결정하는 계산 문제이다.

[0021] [0021] 예를 들어, 상기 문제는 도시들 중 어느 곳도 재방문함이 없이 도시들 각각을 모두 방문하도록 여러 도시들 사이의 루트를 결정해야 하는 유명한 여행 판매원문제일 수 있다. 이 문제에는 여러 가능한 솔루션들-즉, 도시들 간의 경로들이 있다. 여행 판매원 문제와 관련된 최적화 문제는 가능한 많은 경로들 중에서 여행 판매원 문제의 요구 사항들을 충족하는 가장 짧은 경로 - 즉, 가장 좋은 또는 최적의 경로를 찾는 것이다.

[0022] [0022] 상기 문제의 다른 예는 최대 절단 문제(the max cut problem)이다. 그래프에서, 최대 컷 문제의 해결은 꼭지점 집합의 부분집합S와 상보적인 부분 집합(S에 없는 꼭지점의 부분 집합) 사이의 엣지들의 수를 가능한 한 크게 되도록 하는 꼭지점 집합의 부분 집합 S를 의미한다.

[0023] [0023] 해리 곡선(a dissociation curve)의 계산은 변분 문제들의 다른 비-제한적인 예이고, 이는 차원의 수

(dimensionality), 노드들의 수, 원하는 정확도 레벨, 및 기타 요인들에 따라, 어렵거나 쉬울 수 있다(hard or easy). 해리 곡선은, 그래프 상에서, 분자의 원자들을 분리하는 에너지 비용을 표시한다. 그래프의 x-축은 원자들 사이의 거리이고, 그래프의 y-축은 에너지의 양이다.

[0024] [0024] 컴퓨터가 다항 시간에(in polynomial time) 최적의 솔루션을 계산할 수 있도록 컴퓨터 상에서 실행할 변분 문제를 구성하는 것은 그 자체로 어려운 문제이다. 최근까지, 변분 문제들을 실행하는 데 사용할 수 있는 컴퓨팅 자원들은 여기에 설명된 종래 컴퓨터들뿐이었다. 많은 변분 문제들은 종래 컴퓨터들이 합리적인 자원들로 다항 시간에 계산하기에는 너무 어렵거나 너무 복잡하다. 일반적으로, 합리적인 시간에 합리적인 자원들로 계산할 수 있는 근사 솔루션이 그러한 경우들에서 근접-최적 솔루션(the near-optimal solution)으로 받아들여진다.

[0025] [0025] 양자 컴퓨팅의 출현은 변분 문제들의 계산을 포함하여, 컴퓨팅의 많은 영역에서 개선 가능성들을 제공하였다. 양자 컴퓨팅 시스템은 솔루션 공간에서 한 번에 많은 솔루션들을 평가할 수 있기 때문에, 본 발명의 예시적인 실시예들은 그러한 시스템들이 변분 문제들을 처리하는데 특히 적합하다는 것을 인식한다.

[0026] [0026] 변분 양자 알고리즘들(Variational quantum algorithms)은 특정 문제를 해결하는 양자 회로를 위해 고-차원 비-볼록 파라미터 공간(a high-dimensional non-convex parameter space)을 검색하는 고전적 최적화기들(classical optimizers)을 사용한다. 일반적으로 비-볼록 고-차원 공간들에서의 최적화는 매우 어렵고, (의미 있는) 차원의 수가 증가할수록 솔루션에 대한 수렴이 점점 어려워진다. 양자 변분 문제들의 경우, 이것은 현재 기술로 고-차원 비-볼록 공간들에서 최적화를 요구하는 대부분의 문제들을 해결하는 것이, 불가능하지는 않더라도, 매우 어려운 것임을 의미한다. 최적화 화학 또는 복잡한 최적화 문제들은 안정적으로 수렴하기 위해 수천 개의 변분 파라미터들이 필요할 수 있지만, 현대적인 방법은 일반적으로 수십 개의 이러한 파라미터들로 수렴하려고 애쓴다. 본 발명의 주제인 시스템은, 바람직한 실시예들에 따라, 원하는 정확도로의 수렴이, 이전에는 불가능했던 화학 및 최적화 문제들에 대한 수렴의 전망을 개선할 수 있게 했다. 관련 주제들은 해리 곡선 수렴 개선을 위한 부트스트래핑(bootstrapping), 게이트-기반 양자 단열 방법들(gate-based quantum adiabatic methods), 볼록 최적화를 위한 호모토피 방법들(homotopy methods for convex optimization), 최적화를 위한 내부 점 방법들(interior point methods for optimization)이 있다.

### 과제의 해결 수단

[0027] [0027] 본 발명의 예시적 실시 예들은 방법, 시스템, 및 컴퓨터 프로그램 제품을 제공한다. 일 실시 예는 양자 컴퓨팅에서 어려운 변분 양자 문제들(hard variational quantum problems)을 해결하기 위해 중간 재-최적화를 사용하는 단열 진행 방법(a method for adiabatic progression with intermediate re-optimization)을 포함한다. 일 실시 예는 변분 문제(variational problem)에 대응하는 양자 회로를 실행하도록 하이브리드 고전-양자 컴퓨팅 디바이스(a hybrid classical-quantum computing device)를 구성하는 단계를 포함한다. 상기 구성하는 단계는 상기 하이브리드 고전 양자-컴퓨팅 디바이스가 단열 진행 연산(adiabatic progression operation)을 수행함으로써 상기 양자 회로를 실행하게 하는 단계를 포함하고, 상기 단열 진행 연산은 상기 문제의 단순화된 버전으로부터 상기 변분 문제로 상기 변분 문제의 나이도를 증가시키는 단계를 포함한다.

[0028] [0028] 일 실시 예는 컴퓨터 사용 가능 프로그램 제품을 포함한다. 상기 컴퓨터 사용 가능 프로그램 제품은 컴퓨터-판독가능 스토리지 디바이스 및 상기 스토리지 디바이스에 저장된 프로그램 명령들을 포함한다.

[0029] [0029] 일 실시 예는 컴퓨터 시스템을 포함한다. 상기 컴퓨터 시스템은 프로세서, 컴퓨터 판독-가능 메모리, 및 컴퓨터 판독-가능 스토리지 디바이스, 및 상기 메모리를 상기 프로세서에 의한 실행을 위한 상기 스토리지 디바이스에 저장된 프로그램 명령들을 포함한다.

[0030] [0030] 일 실시 예에 따라, 컴퓨터-구현 방법이 제공되고, 상기 컴퓨터-구현 방법은: 변분 문제(variational problem)에 대응하는 양자 회로를 실행하도록 하이브리드 고전-양자 컴퓨팅 디바이스(a hybrid classical-quantum computing device)를 구성하는 단계를 포함하고: 상기 구성하는 단계는: 상기 하이브리드 고전 양자-컴퓨팅 디바이스가 단열 진행 연산(adiabatic progression operation)을 수행함으로써 상기 양자 회로를 실행하게 하는 단계; 상기 변분 문제에 대응하는 파라미터화된 최적화 문제의 연속 파라미터(a continuous parameter of a parametrized optimization problem)에서 제1 증가를 생성하는 단계-상기 연속 파라미터의 현재 상태는 상기 최적화 문제의 해결 가능한 상태(a solvable state)에 대응함-; 상기 제1 증가에 대응하는 한 세트의 변분 파라미터들을 생성하는 단계-; 상기 변분 파라미터들 세트는 상기 양자 회로의 구성을 명시함-; 및 상기 양자 회로의 출력이 상기 문제의 알려진 올바른 솔루션 (a known correct solution of the problem)과 임계량보다 적거나 또는 동일한 양만큼(by less than or equal to a threshold amount) 다르다고 결정하는 것에 응답하여, 상

기 연속 파라미터의 현재 상태를 상기 제1 증가로 대체하는 단계를 포함한다.

[0031] 다른 실시 예에 따라, 컴퓨터 사용 가능 프로그램 제품이 제공되고, 상기 컴퓨터 사용 가능 프로그램 제품은 하나 또는 그 이상의 컴퓨터-판독가능 스토리지 디바이스들, 및 상기 하나 또는 그 이상의 스토리지 디바이스 중 적어도 하나에 저장된 프로그램 명령들 포함하고, 상기 저장된 프로그램 명령들은: 변분 문제 (variational problem)에 대응하는 양자 회로를 실행하도록 하이브리드 고전-양자 컴퓨팅 디바이스(a hybrid classical-quantum computing device)를 구성하는 단계를 수행하는 프로그램 명령들을 포함하고: 상기 구성하는 단계는: 상기 하이브리드 고전 양자-컴퓨팅 디바이스가 단열 진행 연산(adiabatic progression operation)을 수행함으로써 상기 양자 회로를 실행하게 하는 단계; 상기 변분 문제에 대응하는 파라미터화된 최적화 문제의 연속 파라미터(a continuous parameter of a parametrized optimization problem)에서 제1 증가를 생성하는 단계-상기 연속 파라미터의 현재 상태는 상기 최적화 문제의 해결 가능한 상태(a solvable state)에 대응함-; 상기 제1 증가에 대응하는 한 세트의 변분 파라미터들을 생성하는 단계-; 상기 변분 파라미터들 세트는 양자 회로의 구성을 명시함-; 및 상기 양자 회로의 출력이 상기 문제의 알려진 올바른 솔루션(a known correct solution of the problem)과 임계량보다 적거나 또는 동일한 양만큼(by less than or equal to a threshold amount) 다르다고 결정하는 것에 응답하여, 상기 연속 파라미터의 현재 상태를 상기 제1 증가로 대체하는 단계를 포함한다.

[0032] 또 다른 실시 예에 따라, 컴퓨터 시스템이 제공되며, 상기 컴퓨터 시스템은 하나 혹은 그 이상의 프로세서들, 하나 혹은 그 이상의 컴퓨터 판독-가능 메모리들, 하나 혹은 그 이상의 컴퓨터 판독-가능 스토리지 디바이스들 및 상기 하나 혹은 그 이상의 컴퓨터 판독-가능 메모리들 중 적어도 하나를 통해 상기 하나 또는 그 이상의 프로세서들 중 적어도 하나에 의한 실행을 위한 상기 하나 또는 그 이상의 스토리지 디바이스들 중 적어도 하나에 저장된 프로그램 명령들을 포함하고, 상기 저장된 프로그램 명령들은: 변분 문제(variational problem)에 대응하는 양자 회로를 실행하도록 하이브리드 고전-양자 컴퓨팅 디바이스(a hybrid classical-quantum computing device)를 구성하는 단계를 수행하는 프로그램 명령들을 포함하며: 상기 구성하는 단계는: 상기 하이브리드 고전 양자-컴퓨팅 디바이스가 단열 진행 연산(adiabatic progression operation)을 수행함으로써 상기 양자 회로를 실행하게 하는 단계; 상기 변분 문제에 대응하는 파라미터화된 최적화 문제의 연속 파라미터(a continuous parameter of a parametrized optimization problem)에서 제1 증가를 생성하는 단계-상기 연속 파라미터의 현재 상태는 상기 최적화 문제의 해결 가능한 상태(a solvable state)에 대응함-; 상기 제1 증가에 대응하는 한 세트의 변분 파라미터들을 생성하는 단계-; 상기 변분 파라미터들 세트는 양자 회로의 구성을 명시함-; 및 상기 양자 회로의 출력이 상기 문제의 알려진 올바른 솔루션 (a known correct solution of the problem)과 임계량보다 적거나 또는 동일한 양만큼(by less than or equal to a threshold amount) 다르다고 결정하는 것에 응답하여, 상기 연속 파라미터의 현재 상태를 상기 제1 증가로 대체하는 단계를 포함한다.

**도면의 간단한 설명**

[0033] 이제 본 발명의 바람직한 실시예들이 다음 도면을 참조하여, 단지 예로서 설명될 것이다:

도 1은 예시적인 실시예들이 구현될 수 있는 데이터 처리 시스템의 네트워크의 블록도를 도시한다;

도 2는 예시적인 실시예들이 구현될 수 있는 데이터 처리 시스템의 블록도를 도시한다.

도 3은 예시적인 실시예에 따른, 양자 컴퓨팅에서 어려운 변분 양자 문제들을 해결하기 위해 중간 재-최적화를 사용하는 단열 진행을 위한 예시적 구성의 블록도를 도시한다;

도 4는 예시적인 실시예에 따른, 양자 컴퓨팅에서 어려운 변분 양자 문제들을 해결하기 위해 중간 재-최적화를 사용하는 단열 진행(adiabatic progression)을 위한 예시적인 구성의 블록도를 도시한다;

도 5는 예시적인 실시예에 따른, 양자 컴퓨팅에서 어려운 변분 양자 문제를 해결하기 위해 중간 재-최적화를 사용하는 단열 진행을 사용하여 해결하기에 적합한 어려운 변분 양자 문제의 예를 도시한다;

도 6은 예시적인 실시예에 따른, 중간 재-최적화를 사용하는 단열 진행을 사용하여 최대 컷 문제를 해결하는 예를 도시한다;

도 7은 예시적인 실시예에 따른, 중간 재-최적화를 사용하는 단열 진행을 사용하여 해리 곡선을 계산하는 예를 도시한다;

도 8은 예시적인 실시예에 따른, 중간 재-최적화를 사용하는 단열 진행을 사용하여 해리 곡선을 계산하는 다른 예를 도시한다;

도 9는 예시적인 실시예에 따른 양자 컴퓨팅에서 어려운 변분 양자 문제들을 해결하기 위해 중간 재-최적화를 사용하는 단일 진행을 위한 예시적인 프로세스의 플로를 도시한다; 및

도 10은 예시적인 실시예에 따른 양자 컴퓨팅에서 어려운 변분 양자 문제들을 해결하기 위한 중간 재-최적화를 사용하는 단일 진행을 위한 또 다른 예시적 프로세스의 플로를 도시한다.

**발명을 실시하기 위한 구체적인 내용**

[0034] [0034] 변분 문제들의 계산은 노력의 잘-알려진 기술 분야이다. 양자 큐비트들로 형성된 프로세서들을 사용하는 양자 컴퓨팅은 노력의 잘 알려진 또 다른 기술 분야이다. 이들 두 분야들의 노력이 결합된 현재 기술의 상태는 특정 결점들과 한계들을 가지고 있다. 본 발명의 예시적인 실시예들의 동작들 및/또는 구성들은 노력의 이들 기술 분야들에서 종래 기술을 개선하는데 추가적인, 또는 새로운 능력을 제공한다. 특히 양자 컴퓨팅에서 어려운 변분 양자 문제들(hard variational quantum problems)을 해결하기 위해 중간 재-최적화를 사용하여 단일 진행을 하는데 추가적인, 또는 새로운 능력을 제공한다.

[0035] [0035] 본발명의 예시적인 실시예들은, 양자 컴퓨팅에서, 최적화 문제와 같은, 문제를 해결하려면 일반적으로 상기 최적화 문제를, 그 것의 입력들과 함께, 아이싱 해밀턴(Ising Hamiltonian)으로 변환하고(translating), 그런 다음 상기 아이싱 해밀턴을 변분 양자 고유해결사(VQE)알고리즘(Variational Quantum Eigensolver(VQE) algorithm) 및 양자 근사 최적화 알고리즘 Quantum Approximate Optimization Algorithm: QAOA)과 같은, 양자 변분 알고리즘에 전달하는(passing) 것을 필요로 한다는 것을 인식한다.

[0036] [0036] 아이싱(Z. Physik, 31, 253, 1925)은, +1(↑) 및 -1(↓) 값들만 취할 수 있는, "스핀" 변수들  $s_i$ 의 격자로 구성된 모델을 소개했다. 모든 스핀은 외부 자기장  $h$ 뿐만아니라 그 것의 최근접 이웃들(its nearest neighbors)(1D에서 2)과 상호 작용한다. 도 3의 1-차원 아이싱 모델을 참조한다.

[0037] [0037] 예시적 아이싱 모델의 해밀턴은 다음과 같다.

$$H(\{s_i\}) = -J \sum_{\langle i,j \rangle} s_i s_j - h \sum_i s_i$$

[0038] [0038] 합  $\langle i, j \rangle$ 은 전체 최근접 이웃들(1D에서  $j = i \pm 1$ )이다.

[0039] [0039] J는 상호작용의 강도를 명시하는 상수이다. 아이싱 모델에서 "스핀"과 "자기장"이라는 용어는 철과 같은 강자성 재료들의 자발적 자화 현상(the phenomenon of spontaneous magnetization)에 처음 적용한 것에서 유래했다. 각각의 철 원자는 짝을 이루지 않은 전자(an unpaired electron)를 갖고 있고 따라서 순 스핀(a net spin)(또는 자기 모멘트)을 갖는다. 저온에서, 상기 스핀들은 자발적으로 정렬되어 0-이 아닌 거시적 자기 모멘트(a non-zero macroscopic magnetic moment)를 발생시킨다. 상기 거시적 자기 모멘트는 온도가 퀴리 온도(철의 경우 1043K)를 초과하면 사라진다.

[0040] [0040] 아이싱 모델은 이원 합금들의 상 분리(phase separation in binary alloys), 결정 성장, 및 최적화 문제들 해결과 같은, 자성 이외의 많은 다른 문제들에 적용될 수 있다. 더 높은 차원의 아이싱 모델들이 일반적으로 많은 문제들을 해결하는 데 사용된다.

[0041] [0041] 2D 아이싱 모델은 주기적인 경계 조건들 하에서 N 스핀들의 정사각형 격자에 대해 정의된다. 다시, 상기 해밀턴을 기술하면 다음과 같이 쓸 수 있다.

$$H(\{s_i\}) = -J \sum_{\langle i,j \rangle} s_i s_j - h \sum_i s_i$$

[0042] [0042] J는 상호 작용의 강도이고, h는 외부 자기장이며, 합은 전체  $\langle i, j \rangle$  최근접-이웃 쌍들이다. 각각의 스핀은 4개의 최근접 이웃들을 갖는다.

[0043] [0043] 최적화 및 기타 변분 문제들은 계산-집약적인 작업들이다. 최적화 또는 기타 변분 문제의 솔루션 공간에 대한 철저한 분석은 상업적으로 이용 가능한 종래 컴퓨터에서 쉽게 몇 년이 걸릴 수 있다. 따라서, 본 발명의 예시적인 실시예들은 또한 양자 컴퓨터들을 사용하여 시간-효율적인 방식으로 최적의 솔루션을 계산하는 것이 현재 이용 가능한 방법들을 사용하여서는 훨씬 더 어렵다는 것을 인식한다.

- [0046] [0044] 본 발명의 예시적인 실시예들은 원하는 정도의 정확도로 어려운 변분 문제(a hard variational problem)를 해결하는데, 사용자의 정확도로 해결될 수 있는, 문제의 단순화된 버전으로 시작하여, 원하는 문제가 최종적으로 해결될 때까지, 각각의 더 복잡한 문제를 점차로 해결하면서 문제의 복잡성을 주의 깊게 증가시켜서, 해결한다. 각각의 더 복잡한 문제를 점차로 해결하면서, 문제의 복잡성을 주의 깊게 증가시키는 것을 여기에서 단열 진행(adiabatic progression)이라고 한다. 예를 들어, 화학의 해리 곡선들의 경우, 이는 해리 곡선의 쉬운 부분에서 시작하여 곡선의 더 어렵고 더 여기된 부분들(harder, more excited sections)로 해밀턴을 진행하는 것을 의미할 수 있다. 또 다른 예로, 타겟 시스템이 그래프로 표현되는 최적화 문제들에서, 그 것은 문제에 대한 소수의 노드들 또는 컴포넌트들로 시작하여 점차적으로 특이점들(singular points)의 수들을 늘리는 것을 의미할 수 있다.
- [0047] [0045] 결정적으로, 예시적인 실시예들은, 여기서 주요 도전인, 문제의 난이도를 진행하는, 속도를 컨트롤하는 방법들을 제공한다(문제를 직접 해결하려는 시도는 푸어 페이스잉(poor pacing)- 쉬운 문제에서 전체 문제로 한번에 점프하는 것으로 보여 질 수 있다). 만일 단계가 너무 빠르게 진행된다면, 중간 최적화의 시작점이, 원하는 솔루션을 포함하는, 매력의 영역을 벗어나서, 유틸리티저가 솔루션을 찾을 가능성을 훨씬 감소시키거나 또는 수렴하는 것이 훨씬 더 오래 걸릴 수 있게 한다.
- [0048] [0046] 예시적인 실시예는 중간 솔루션들이 사용자가 원하는 수준의 정확도에 도달하도록 보장하기 위해 이 페이스를 컨트롤한다. 현재 기술에서 몇 가지 예시적인 문제점들과 종래 기술에 대한 본 발명의 예시적인 실시예들의 개선점들은 다음과 같다. 첫째, 예시적인 실시예들의 방법은 이전에는 변분 알고리즘으로 직접 또는 "부트스트래핑(bootstrapping)"(창 깊이가 1과 동일한 외삽법 (extrapolation)의 형식)을 통해 해결하는 것이 불가능했던 정확도로 해리 곡선들 및 최대 절단 문제들을 완료하기 위해 아래의 "리와인드(rewind)" 방법과 함께 사용된다. 둘째, 고전적인 호모토피(homotopy) 또는 내점(interior-point) 방법을 사용하는 것보다 개선된 점은 일 실시 예가 양자 솔루션을 사용할 수 있다는 것 - 중간 솔루션들의 과도 함수들이 문제의 크기에 비해 기하급수적으로 많은 확률들을 유지할 수 있다는 것이다. 마지막으로, 가까운 장래에, 양자 하드웨어는 충분히 커질 것이고 양자 알고리즘들의 솔루션들을 고전적으로 검증할 수 없을 것이며, 따라서 양자 계산들의 신뢰성을 높이는 문제가 시급하게 될 것이다.
- [0049] [0047] 예시적인 실시예들은 현재 이용 가능한 도구들 또는 솔루션들이 이들 필요사항들(needs)을 해결하지 못함을 또는 이들 필요사항들에 대한 적절한 솔루션들을 제공하지 못함을 인식한다. 본 발명을 설명하기 위해 사용된 예시적인 실시예들은 전술한 문제들 및 기타 관련 문제들을 일반적으로 어드레스하고 해결하는데, 양자 컴퓨팅에서 어려운 변분 양자 문제들을 해결하는 중간 재-최적화를 사용하는 단열 진행에 의해서 해결한다.
- [0050] [0048] 일 실시예는 소프트웨어 애플리케이션으로 구현될 수 있다. 일 실시예, 또는 하나 혹은 그 이상의 관련 컴포넌트들을 구현하는 상기 애플리케이션은, 종래 양자-고전 하이브리드 데이터 처리 시스템의 수정- 즉, 양자 컴퓨팅 시스템에 대한 입력들을 생성하는 고전적 컴퓨팅 시스템의 네이티브 애플리케이션으로서, 네트워크를 통해 종래 양자 컴퓨팅 시스템과 통신하는 고전적 데이터 처리 시스템에서 실행되는 애플리케이션으로서, 다른 방식들로 종래 양자-고전 시스템과 함께 작동하는 별도의 애플리케이션으로서, 고전 시스템 또는 그들의 일부 조합에서 실행하기 위한 스탠드얼론 애플리케이션으로서 구성될 수 있다.
- [0051] [0049] 일 실시예는 고전적 컴퓨팅 환경과 양자 컴퓨팅 환경을 포함하는 하이브리드 데이터 처리 환경을 구성한다. 본 발명의 예시적인 실시예들에 따르면, 상기 환경은 적어도 하나의 양자 컴퓨팅 노드(QCN), 및 일 실시예가 실행할 수 있는 적어도 하나의 종래 노드(CN)를 포함한다. 이러한 컴퓨팅 환경은 이하에서 양자 컴퓨팅 환경(Quantum Computing Environment, QCE)으로 지칭된다. QCE는 - 클러스터와 같은 - 적절한 구성에 하나 혹은 그 이상의 CN을 포함하여 종래 이진 컴퓨팅을 사용하여 애플리케이션을 실행할 수 있다. 하이브리드 환경은, 클라우드 컴퓨팅 아키텍처를 사용하여 구현할 수 있지만 반드시 그런 것은 아니다. 주어진 문제를 해결하기 위해, 일 실시예는 고전적 컴퓨팅 환경에서, 양자 컴퓨팅 환경을 사용하여 실행되는 양자 회로를 생성한다.
- [0052] [0050] 일 실시예는 다음과 같이 작동한다: 단일 연속 파라미터  $z$ 를 갖고,  $z = 0$ 에 대응하는 문제가 해결하기 쉽다는 속성을 갖는, 파라미터화된 문제가 제공될 때,  $z = 1$ (또는 좀 더 높은, 사용자 명시  $z$ )에 대응하는 문제가 실시예가 해결해야 하는 타겟(잠재적으로 어려운) 문제이다. 예를 들어, 만일 해결해야 할 문제가 그래프 상의 최적화(예: 최대 절단)라면, 파라미터화된 문제는 그래프의 엣지들의 수를 컨트롤할 수 있고, 여기서  $z = 0$ 일 때 그래프는 하나의 엣지를 가지며 (문제가 풀기 쉬움),  $z = 1$ 일 때 그래프는 원래 문제 인스턴스(instance)에서 모든 엣지들을 갖는다. 일 실시예에서, 쉬운 문제( $z = 0$ )에서 어려운 문제( $z = 1$ )로의 원활한 진행이 가능하다(이것은 호모토피 최적화 및 내부 포인트 방법들의 분야들에서 자세히 연구됨).

- [0053] [0051] 일 실시 예는 알려진 쉬운 문제로 시작한다. 상기 쉬운 문제는 대응하는 연속 파라미터  $z$  (a corresponding continuous parameter  $z$ )를 갖는다. 상기 연속 파라미터  $z$ 는 한 세트의 대응하는 변분 파라미터들(a corresponding set of variational parameters)을 갖는다. 상기 변분 파라미터들 세트는 상기 문제를 해결하는 데 사용할 수 있는 양자 회로의 구성을 정의한다. 따라서, 문제의 복잡성을 증가시키기 위해, 일 실시예는 연속 파라미터  $z$ 를 일정 양만큼 증가시키고, 이에 따라 변분 파라미터들 세트를 상응하게 변경한다. 변경된 변분 파라미터들 세트는, 보다 복잡한-문제를 해결하는 데 사용할 수 있는, 양자 회로의 새로운 구성을 정의한다.
- [0054] [0052] 일 실시예는 복잡성 증가를 위해, 리와인드 실시예(a rewind embodiment) 및 레이스 실시예(a race embodiment) 중 하나 또는 모두를 포함하는, 많은 방법들을 따를 수 있다. 각 방법은 복잡도 증가를 다르게 수행한다.
- [0055] [0053] 리와인드 실시예는 문제에 대한 알려진 올바른 솔루션이 다른 수단들에 의해서도 획득될 수 있을 때 사용하기에 적합하다. 리와인드 실시예는 쉬운 상태에서 연속 파라미터  $z$ 로 시작하여  $z$ 를 목표 상태(즉, 최대  $z$ )로 증가시킨다. 리와인드 실시예는 새로운  $z$ 에 대응하는 한 세트의 변분 파라미터들을 생성한다. 그런 다음 리와인드 실시예는 문제 해결을 시도하기 위해 상기 변분 파라미터들 세트에 의해 정의된 양자 회로 구성을 사용한다. 리와인드 실시예는 양자 회로의 출력을 다른 수단들(예: 솔루션의 다항-시간 검증의 일부 형태를 사용하거나, 정확한 고전 솔루션과 비교함으로써)을 통해 얻은 문제에 대한 알려진 올바른 솔루션과 비교한다. 만일 양자 회로의 출력이 알려진-올바른 솔루션과 임계값 차이 미만이라면, 실시예는 상기 어려운 문제를 해결하여, 목표를 달성한 것이다. 대신에, 만일 양자 회로의 출력이 알려진-올바른 솔루션과 임계값 차이보다 높거나, 또는 양자 회로가 실행을 완료하지 못하고 미리 결정된 시간 기간 내에 솔루션을 생성하지 못한다면, 리와인드 실시예는  $z$ 의 증가가 너무 컸다는 결론을 내린다.
- [0056] [0054] 만일  $z$ 의 증가가 너무 컸다면, 리와인드 실시예는 시작점에서  $z$ 의 더 작은 증가를 선택한다. 일 실시예에서,  $z$ 의 더 작은 증가는 이전 증가 크기의 절반이다. 다른 실시예에서,  $z$ 의 더 작은 증가는  $1/3$ 이다. 다른 실시예에서, 더 작은 증가는  $1/n$ 에 이전 증가의 크기를 곱한 것이며, 여기서  $n$ 은 1보다 큰 임의의 양의 정수이다. 다른 실시예에서, 더 작은 증가는 다른 적절한 방법을 사용하여 선택될 수 있다.
- [0057] [0055] 일단 리와인드 실시예가 더 작은 증가를 결정하면, 상기 실시예는 새로운  $z$ 에 대응하는 한 세트의 변분 파라미터들을 생성한 후, 문제를 해결하기 위해 새로운 변분 파라미터들 세트에 의해 정의된 양자 회로 구성을 사용하는 프로세스를 반복한다. 새로운  $z$ 가 문제 해결에 성공하면, 상기 실시예는 새로운  $z$ 를 시작점으로 만들고, 여기에 설명된 방식으로  $z$ 를 증가시키려고 시도한다.
- [0058] [0056] 만일 새로운  $z$ 가 문제를 해결하는 데 성공하지 못한다면, 리와인드 실시예는 시작점에서  $z$ 의 더 작은 증가를 선택한다. 일 실시예에서, 더 작은 증가는  $1/n$ 에 이전 증가의 크기를 곱한 것이며, 여기서  $n$ 은 1보다 큰 임의의 양의 정수이다. 다른 실시예에서, 더 작은 증가는 다른 적절한 방법을 사용하여 선택될 수 있다. 리와인드 실시예가 더 작은 증가를 결정하면, 실시예는 새로운  $z$ 에 대응하는 한 세트의 변분 파라미터를 생성한 다음, 문제를 해결하기 위해 새로운 변분 파라미터 세트에 의해 정의된 양자 회로 구성을 사용하는 프로세스를 반복한다.
- [0059] [0057] 리와인드 실시예는 문제를 해결하는 시작점에서  $z$ 의 증가에 도달할 때까지 프로세스를 반복한다. 상기 실시예는 새로운  $z$ 를 시작점으로 만들고 여기에 설명된 방식으로  $z$ 를 다시 한 번 증가시키려고 시도한다. 따라서, 리와인드 실시예는 상기 실시예가  $z$ 의 목표 값을 갖는 올바른 양자 회로 구성을 생성할 때까지 각 단계에서  $z$ 를 가능한 한 많이 증가시키도록 진행한다.
- [0060] [0058] 따라서, 만일 각 증가가 이전 증가보다 작으면, 리와인드 실시예의 연산들에게는 선행 검색의 변분 파라미터들(the predecessor's variational parameters)을 사용할 때 관심 유역 내부에(inside the basin of attraction) 있는 다음 지점에 대해 검색(예를 들어, 각 증가가 이전 증가의 절반일 때, 이전 검색)을 수행하는 것으로 이해될 수 있다. 이 검색은 진전 없는 무한 리와인딩 가능성(the possibility of rewinding infinitely and never making progress)에 의해 제한되므로, 실제 검색에서는 선행 검색에 매우 가까운 어떤 지점들을 보거나 사용하는 것을 선택하지 않는 것이 바람직하다.
- [0061] [0059] 만일 검증 방법이 이용 가능하지 않다면 레이스 실시예를 사용하는 것이 적합하다. 레이스 실시예는 여러 가능한 후속 지점들 중 하나(one of several possible successor points)를 선택하기 위해 하나 혹은 그 이상의 의사-검증 방법(one or more pseudo-validation methods)을 사용한다. 예를 들어, 일 레이스 실시예에서,

관심 유역 내부 대 외부일 때(when inside vs. outside the basin of attraction) 구배-기반 옵티마이저로(with a gradient-based optimizer) 최적 지점에 수렴하는 시간의 차이는 진행 상의 의사-조건(as a pseudo-condition on progressing)으로 사용된다. 이 것은, 만일 상기 실시예가 관심 유역 내부의 지점들이 매우 빠르게 수렴하고, 관심유역 외부의 지점들이 10배 더 오래 걸린다고 결정하면, 상기 실시예는 다음 지점을 선택하기 위해 이들 지점들의 실행을 서로에 대해 "레이스(race)"할 수 있음을 의미한다.

- [0062] [0060] 따라서, 레이스 실시예는 쉬운 상태의 연속 파라미터  $z$ 로 시작하고, 시작지점으로부터  $z$ 에서 한세트의 증가들을 생성한다. 일 실시예에서, 상기 증가들 세트 각각은 이전 증가로부터 고정된 거리이며, 이는 시작점에서 목표 상태까지의 범위를 동일한-크기의 세그먼트들로 분할한다. 다른 실시예는 동일한-크기의 세그먼트들을 사용하지만, 목표 상태 대신 중간 상태까지만 사용한다. 다른 실시예는 문제의 해결 가능한 상태에 도달할 기회를 최대화하기 위해 해결될 문제의 알려진 특성에 기초하여 선택된, 동일하지 않은-크기의 세그먼트들을 사용한다. 예를 들어, 상기 증가는  $z$ 의 한 값 근처에서는 더 작고  $z$ 의 다른 값 근처에서는 더 클 수 있다. 다른 실시예는 상기 증가들 세트를 생성하는 다른 적절한 방법을 사용한다.
- [0063] [0061] 상기 세트에서 각각의 새로운  $z$ 에 대해, 레이스 실시예는 각각의 새로운  $z$ 에 대응하는 변분 파라미터들의 세트를 생성한다. 그런 다음 레이스 실시예는 문제 해결을 시도하기 위해 변분 파라미터들 세트 각각에 의해 정의된 양자 회로 구성을 사용한다.
- [0064] [0062] 양자 회로 구성들의 각각은 병렬로 실행된다. 양자 회로들 세트의 실행 시간들은 클러스터되는(cluster) 경향이 있기 때문에, 레이스 실시예는 제1 양자 회로가 실행을 완료하는 시간을 기록한다. 그 때부터, 일 실시예는 상기 세트의 다른 회로가 실행을 완료할 때까지 추가의 시간 기간을 기다린다. 일 실시예에서, 상기 추가의 시간 기간은 상기 제1 회로의 완료까지의 시간과 동일하다. 다른 실시예에서, 상기 추가의 시간 기간은 제1 회로의 완료까지의 시간의 2배이다. 일 실시예에서, 상기 추가의 시간 기간은 제1 회로의 완료까지의 시간의 3배이다. 다른 실시예들은 제1 회로의 완료까지의 시간의 5배 또는 10배와 같은 조정 가능한 시간 기간, 또는 절대 시간 기간(예를 들어, 10초 또는 1분)을 사용하도록 구성 가능하다.
- [0065] [0063] 일 실시예는 추가의 시간 기간 내에 실행을 완료하는 모든 회로를 문제에 대한 잠재적인 솔루션으로 간주한다. 추가의 시간 기간의 끝에서도 여전히 실행 중인 모든 회로들은 문제에 대한 잠재적인 솔루션들이 아니기 때문에, 일 실시예는 이들 회로들의 실행을 중지한다. 일 실시예는 솔루션들 그룹에서 하나의 회로에 대응하는  $z$ 의 가장 큰 값을 선택한다. 상기 실시예는 상기 새로운  $z$ 를 시작지점으로 만들고, 여기에 설명된 방식으로  $z$ 를 다시 한 번 증가시키려고 시도한다. 따라서, 레이스 실시예는 실시예가 목표값  $z$ 에 도달할 때까지 각 단계에서  $z$ 를 최대한 증가시킨다.
- [0066] [0064] 의사-검증 방법의 다른 실시예는 후보 지점들 각각과 그들의 외삽 파라미터들 각각(their respective extrapolated parameters)에 대해 계산된 변분 비용 함수의 비교(comparison of the computed variational cost function)를 사용한다.
- [0067] [0065] 일 실시예가 연속 파라미터  $z$ 의 목표 값에 도달하여, 어려운 문제를 해결했을 때, 상기 실시예는 실행을 종료하고, 문제 솔루션을 반환한다. 만일 일 실시예가  $z$ 의 목표 값에 도달하는 데 실패하면, 상기 실시예는 도달 가능했던  $z$ 의 값과, 상기 문제의 복잡성의 대응 레벨에 대한 솔루션을 반환한다.
- [0068] [0066] 본 명세서에서 설명된 일 실시예는 단열 진행(adiabatic progression)을 사용하여 화학, 머신 러닝, 및 최적화 문제를 해결하는 방법을 제공하지만, 상기 방법에서, 노이즈 중간 규모 양자(Noisy Intermediate Scale Quantum: NISQ) 디바이스들에 대해 QAOA 또는 단열 시뮬레이션을 지지할 수 없게(untenable) 만드는 값비싼 트로터화 또는 해밀턴 지수화 단계들(the expensive trotterization or hamiltonian exponentiation steps)을 수행할 필요는 없다. 일 실시예는 NISQ 디바이스에 최적으로 적합한, 매우 정사각형인 회로들을 생성한다. 일 실시예는 문제의 난이도를 진행하도록 속도를 정확하게 컨트롤한다.
- [0069] [0067] 본 명세서에 기술된 양자 컴퓨팅에서 어려운 변분 양자 문제를 해결하기 위해 중간 재-최적화를 사용하는 본 발명의 단열 진행 방식은 양자 컴퓨팅에 관련된 기술분야, 특히 어려운 변분 문제를 해결하기 위해 양자 데이터 처리 환경을 운영하는 것과 관련된 기술 분야에서 현재 이용 가능한 방법들에서는 찾아볼 수 없다. 본 명세서에 설명된 실시예의 방법은, 디바이스 또는 데이터 처리 시스템에서 실행하도록 구현될 때, 양자 컴퓨팅에서 어려운 변분 양자 문제들을 해결하기 위해 중간 재-최적화를 사용하는 단열 진행에서, 상기 디바이스 또는 데이터 처리 시스템의 기능의 실질적 향상을 포함한다.
- [0070] [0068] 예시적인 실시예들은 단지 예들로서 특정 유형의 알고리즘들, 라이브러리들, 코드, 명령들, 차원들, 데

이터, 디바이스들, 데이터 처리 시스템들, 환경들, 컴포넌트들, 및 애플리케이션들과 관련하여 설명된다. 이들 및 기타 유사한 실시 예들의 특정 표현은 본 발명을 제한하도록 의도되지 않는다. 이들 및 다른 유사한 실시 예들의 임의의 적절한 표현은 예시적인 실시예들의 범위 내에서 선택될 수 있다.

- [0071] [0069] 또한, 예시적인 실시예들은 임의의 유형의 데이터, 데이터 소스, 또는 데이터 네트워크를 통한 데이터 소스에 대한 액세스와 관련하여 구현될 수 있다. 임의의 유형의 데이터 스토리지 디바이스는, 본 발명의 범위 내에서 데이터 처리 시스템에서 로컬로 또는 데이터 네트워크를 통해, 본 발명의 실시예에 데이터를 제공할 수 있다. 실시예가 모바일 디바이스를 사용하여 설명되는 경우, 모바일 디바이스와 함께 사용하기에 적합한 모든 유형의 데이터 스토리지 디바이스는 예시적인 실시예의 범위 내에서, 모바일 디바이스에서 로컬로 또는 데이터 네트워크를 통해, 이러한 실시예들에 데이터를 제공할 수 있다.
- [0072] [0070] 예시적인 실시예들은 단지 예들로서 특정 코드, 디자인들, 아키텍처들, 프로토콜들, 레이아웃들, 개략도들, 및 도구들을 사용하여 설명되며 예시적인 실시예들로 제한되지 않는다. 또한, 예시적인 실시예들은 설명의 명확성을 위한 예들로서만 특정 소프트웨어, 도구들, 및 데이터 처리 환경들을 사용하여 일부 경우에 설명된다. 예시적인 실시예는 다른 비교 가능하거나 유사한 목적의 구조들, 시스템들, 애플리케이션들 또는 아키텍처들과 함께 사용될 수 있다. 예를 들어, 다른 비교 가능한 모바일 디바이스들, 구조들, 시스템들, 애플리케이션들, 또는 아키텍처들이, 본 발명의 범위 내에서 본 발명의 이러한 실시예와 함께 사용될 수 있다. 예시적인 실시예는 하드웨어, 소프트웨어, 또는 이들의 조합으로 구현될 수 있다.
- [0073] [0071] 본 명세서의 예는 설명의 명료성을 위해서만 사용되며 예시적인 실시예에 제한되지 않는다. 추가 데이터, 연산들(operations), 동작들(actions), 과제들, 활동들, 및 조작들은 본 명세서로부터 생각해 낼 수 있고, 이들은 예시적인 실시예들의 범위 내에 포함된다.
- [0074] [0072] 여기에 열거된 모든 이점들은 단지 예들일 뿐이며 예시적인 실시예들로 제한되는 것으로 의도되지 않는다. 특정한 예시적인 실시예들에 의해 추가적인 또는 다른 이점들이 실현될 수 있다. 게다가, 특정한 예시적인 실시예는 위에 열거된 이점들 중 일부, 전부, 또는 전혀 갖지 않을 수 있다.
- [0075] [0073] 도면을 참조하고 특히 도 1 및 2를 참조하면, 이들 도면들은 예시적인 실시예들이 구현될 수 있는 데이터 처리 환경들의 예시적인 다이어그램들이다. 도 1 및 도 2는 단지 예시들일 뿐이고, 다른 실시예가 구현될 수 있는 환경들과 관련하여 임의의 제한을 주장하거나 암시하도록 의도되지 않는다. 특정 구현은 다음 설명을 기반으로 묘사된 환경들에 대해 많은 수정들을 할 수 있다.
- [0076] [0074] 도 1은 예시적인 실시예들이 구현될 수 있는 데이터 처리 시스템들의 네트워크의 블록도를 도시한다. 데이터 처리 환경(100)은 예시적인 실시예들이 구현될 수 있는 고전적인 컴퓨터들의 네트워크다. 데이터 처리 환경(100)은 네트워크(102)를 포함한다. 네트워크(102)은 데이터 처리 환경(100) 내에서 함께 연결된 다양한 디바이스들과 컴퓨터들 사이의 통신 링크들을 제공하는 데 사용되는 매체이다. 네트워크(102)은 유선, 무선 통신 링크들 또는 광섬유 케이블들과 같은 연결들을 포함할 수 있다.
- [0077] [0075] 클라이언트들 또는 서버들은 네트워크(102)에 연결된 특정 데이터 처리 시스템들의 역할들의 예일뿐이며 이러한 데이터 처리 시스템들에 대한 다른 구성들 또는 역할들을 배제하려는 것은 아니다. 서버(104) 및 서버(106)는 고전적인 데이터 처리 시스템들이고 스토리지 유닛(108)과 함께 네트워크(102)에 연결된다. 소프트웨어 애플리케이션들은 데이터 처리 환경(100)의 임의의 컴퓨터 상에서 실행할 수 있다. 클라이언트들(110, 112, 114)도 또한 네트워크(102)에 연결된다. 서버(104 또는 106) 또는 클라이언트(110, 112 또는 114)와 같은, 데이터 처리 시스템은, 데이터를 포함할 수 있고 그 위에서 실행되는 소프트웨어 애플리케이션들 또는 소프트웨어 도구들을 가질 수 있다.
- [0078] [0076] 단지 예로서, 그리고 그러한 아키텍처에 대한 어떠한 제한도 암시하지 않고, 도 1은 일 실시예의 예시적인 구현에서 사용가능한 특정 컴포넌트들을 도시한다. 예를 들어, 서버들(104 및 106), 및 클라이언트들(110, 112, 114)은, 단지 예들로서 서버들 및 클라이언트들로서 도시되며 클라이언트-서버 아키텍처에 대한 제한을 의미하지 않는다. 다른 예로서, 일 실시예는 도시된 바와 같이 여러 데이터 처리 시스템들 및 데이터 네트워크에 걸쳐 분산될 수 있는 반면, 다른 실시예는 예시적인 실시예들의 범위 내에서 단일 데이터 처리 시스템에서 구현될 수 있다. 데이터 처리 시스템들(104, 106, 110, 112, 및 114)은 또한 클러스터의 예시적인 노드들, 파티션들, 및 실시예를 구현하기에 적합한 다른 구성들을 나타낸다.
- [0079] [0077] 디바이스(132)는 여기에 설명된 디바이스의 예이다. 예를 들어, 디바이스(132)는 스마트폰, 태블릿 컴퓨터, 랩톱 컴퓨터, 고정 또는 휴대용 형태의 클라이언트(110), 웨어러블 컴퓨팅 디바이스, 또는 임의의 다른 적

절한 디바이스의 형태를 취할 수 있다. 도 1에서 다른 데이터 처리 시스템에서 실행되는 것으로 설명된 임의의 소프트웨어 애플리케이션은 유사한 방식으로 디바이스(132)에서 실행되도록 구성될 수 있다. 도 1의 다른 데이터 처리 시스템에 저장되거나 생성된 임의의 데이터 또는 정보는 유사한 방식으로 디바이스(132)에 저장되거나 생성되도록 구성될 수 있다.

[0080] [0078] QCE(140)는 여기에 설명된 QCE의 예이다. 예를 들어, QCE(140)는 CN(104, 106), 및 많은 다른 유사한 CN(142)을 포함한다. 예로써, CN(106 및 142)은 CN의 클러스터(144)로서 구성될 수 있다. QCE(140)는 QCN(146)과 같은 하나 혹은 그 이상의 QCN들을 더 포함한다. QCN(146)과 같은 QCN은, 하나 이상의 q-프로세서들(148)를 포함한다. 현재 실행 가능한 큐비트는 q-프로세서(148)의 예이다. 애플리케이션(105)은 여기에 설명된 실시 예를 구현한다. 애플리케이션(105)은, QCE(140)의 서버(104)와 같은 CN 상에서 동작한다. 애플리케이션(105)은 연산 라이브러리, 회로들, 및 스토리지(108) 내의 메타데이터, 또는 임의의 다른 적절한 스토리지에 저장한다.

[0081] [0079] QCE(140)는 유선 연결들, 무선 통신 프로토콜들, 또는 다른 적절한 데이터 연결을 사용하여 네트워크(102)에 연결될 수 있다. 클라이언트들(110, 112, 114)는, 예를 들어, 개인용 컴퓨터들 또는 네트워크 컴퓨터들일 수 있다. 네트워크(102)는 TCP/IP(Transmission Control Protocol/Internet Protocol) 및 다른 프로토콜들을 사용하여 서로 통신하는 네트워크 및 게이트웨이들의 집합을 나타낼 수 있다. 도 1은 예로서 의도된 것이며, 다른 예시적인 실시예들에 대한 아키텍처 제한으로 의도된 것이 아니다.

[0082] [0080] 데이터 처리 환경(100)은 또한 클라우드의 형태를 취할 수 있고, 편리한 서비스 제공을 위해 클라우드 컴퓨팅 모델을 사용하며, 최소한의 관리 노력 또는 서비스 제공자와의 상호 작용으로 신속하게 제공 및 해제할 수 있는 구성 가능한 컴퓨팅 자원(예: 네트워크, 네트워크 대역폭, 서버, 처리, 메모리, 스토리지, 애플리케이션, 가상 머신 및 서비스)의 공유 풀에 대한 주문형 네트워크 액세스이다.

[0083] [0081] 도 2를 참조하면, 이 도면은 예시적 실시 예들이 구현될 수 있는 데이터 처리 시스템의 블록 다이어그램을 도시한다. 데이터 처리 시스템(200)은 도 1의 서버들(104 및 106) 또는 클라이언트들(110, 112, 114)와 같은 컴퓨터, 또는 본 발명의 예시적 실시 예들을 위해 프로세스들을 구현하는 컴퓨터 사용 가능 프로그램 코드 또는 명령들이 위치될 수 있는 다른 유형의 디바이스의 예이다.

[0084] [0082] 데이터 처리 시스템(200)은, 본발명의 예시적 실시 예들의 프로세스들을 구현하는 컴퓨터 사용 가능 프로그램 코드 또는 명령들 위치될 수 있는 도 1의 데이터 처리 시스템(132)와 같은, 데이터 처리 시스템 또는 그 안의 구성을 또한 나타낸다. 데이터 처리 시스템(200)은 오직 하나의 예로서만, 컴퓨터로 기술되며, 이에 국한되지는 않는다. 도 1의 디바이스(132)와 같은, 다른 디바이스들의 형태의 구현들은, 여기서 설명한 데이터 처리 시스템(200)의 동작들 및 기능들의 일반적인 설명으로부터 벗어남이 없이, 데이터 처리 시스템(200)에 터치 인터페이스를 추가함과 같은 방식에 의해서 데이터 처리 시스템(200)을 수정할 수도 있고, 데이터 처리 시스템(200)으로 부터 특성의 도시된 컴포넌트들을 제거할 수도 있다.

[0085] [0083] 상기 도시된 예에서 데이터 처리 시스템(200)은 노스 브리지(North Bridge) 및 메모리 컨트롤러 허브(NB/MCH)(202)와 사우스 브리지(South Bridge) 및 I/O(입출력) 컨트롤러 허브(SB/ICH)(204)를 포함하는 허브 아키텍처를 채용한다. 처리 유닛(206), 메인 메모리(208), 그래픽 프로세서(210)은 노스 브리지 및 메모리 컨트롤러 허브(NB/MCH)(202)에 결합된다. 처리 유닛(206)은 하나 또는 그 이상의 프로세서들을 포함할 수 있으며, 하나 또는 그 이상의 이기종 프로세서 시스템들을 사용하여 구현될 수도 있다. 처리 유닛(206)은 멀티-코어 프로세서일 수 있다. 그래픽 프로세서(210)은 특정 구현들에서 가속 그래픽 포트(AGP)를 통해 NB/MCH(202)에 결합될 수 있다.

[0086] [0084] 상기 도시된 예에서, 근거리 네트워크(LAN) 어댑터(212)는 사우스 브리지(South Bridge) 및 I/O컨트롤러 허브(I/O Controller Hub)(SB/ICH)(204)에 결합된다. 오디오 어댑터(216), 키보드 및 마우스 어댑터(220), 모뎀(222), 판독 전용 메모리(ROM)(224), 범용 직렬 버스(USB) 및 기타 포트(232) 및 PCI/PCIe 디바이스들(234)은 사우스 브리지 및 I/O 컨트롤러 허브(204)에 버스(238)를 통해 결합된다. 하드 디스크 드라이브(HDD) 또는 솔리드 스테이트 드라이브(SSD)(226) 및 CD-ROM(230)은 버스(240)를 통해 사우스 브리지 및 I/O 컨트롤러 허브(204)에 결합된다. PCI/PCIe 디바이스들(234)은, 예를 들어, 이더넷 어댑터들, 애드-인 카드들(add-in cards), 및 노트북 컴퓨터들을 위한 PC 카드들을 포함할 수 있다. PCI는 카드 버스 컨트롤러를 사용하지만, PCIe는 사용하지 않는다. ROM(224)는, 예를 들어, 플래시 이진 입출력 시스템(BIOS)일 수 있다. 하드 디스크 드라이브(226) 및 CD-ROM(230)은, 예를 들어, 통합 드라이브 전자 장치(IDE), 직렬 고급 기술 부착 장치(SATA) 인터페이스, 또는 외장-SATA(eSATA) 및 마이크로-SATA(micro-SATA)와 같은 변종들을 사용할 수 있다. 슈퍼(Super) I/O(SIO) 디바이스(236)은 버스(238)를 통해 사우스 브리지 및 I/O 컨트롤러 허브(SB/ICH)(204)에 결합될 수 있다.

- [0087] [0085] 메인 메모리(208), ROM(224) 또는 플래시 메모리(도시되지 않음)와 같은, 메모리들은 컴퓨터 사용 가능 스토리지 디바이스들의 일부 예들이다. 하드 디스크 드라이브 또는 솔리드 스테이트 드라이브(226), CD-ROM(230) 및 기타 유사한 사용 가능 디바이스들도 컴퓨터 사용 가능 스토리지 매체를 포함한 컴퓨터 사용 가능 스토리지 디바이스들의 일부 예들이다.
- [0088] [0086] 운영 체제는 처리 유닛(206)에서 실행된다. 운영 체제는 도 2의 데이터 처리 시스템(200) 내에서 다양한 컴포넌트들을 조정하고 컨트롤한다. 운영 체제는 서버 시스템들, 개인용 컴퓨터들 및 모바일 디바이스들을 포함 하되 이에 국한되지 않는 모든 유형의 컴퓨팅 플랫폼을 위한 상업적으로 사용할 수 있는 운영 체제일 수 있다. 객체 지향 또는 다른 유형의 프로그래밍 시스템이 상기 운영 체제와 함께 운영될 수 있으며 데이터 처리 시스템 (200)에서 실행되는 프로그램들 또는 애플리케이션 들로부터 상기 운영 체제에 호출들을 제공할 수 있다.
- [0089] [0087] 운영 체제, 객체 지향 프로그래밍 시스템 및 도 1의 애플리케이션(105)과 같은 애플리케이션들 또는 프 로그램들을 위한 명령들은 하드 디스크 드라이브(226) 상의 코드(226A) 형태로, 스토리지 디바이스들 상에 위치 될 수 있고, 처리 유닛(206)에 의한 실행을 위해, 메인 메모리(208)과 같은, 하나 또는 그 이상의 메모리들 중 적어도 하나에 로드 될 수 있다. 예시적 실시 예들의 프로세스들은 컴퓨터 구현 명령들을 사용하여 처리 유닛 (206)에 의해서 수행될 수 있고, 상기 컴퓨터 구현 명령들은, 예를 들어, 메인 메모리(208), 판독 전용 메모리 (224) 또는 하나 또는 그 이상의 주변 장치들에 위치할 수 있다.
- [0090] [0088] 또한, 하나의 경우에서, 코드(226A)는 원격 시스템(201B)으로부터 네트워크(201A)을 통해 다운로드 될 수 있으며, 유사한 코드(201C)는 원격 시스템(201B)의 스토리지 디바이스(201D)에 저장되어 있다. 다른 경우에 서는, 코드(226A)가 네트워크(201A)를 통해 원격 시스템(201B)으로 다운로드 될 수 있으며, 다운로드 된 코드 (201C)는 스토리지 디바이스 (201D)에 저장된다.
- [0091] [0089] 도 1-2의 하드웨어는 구현에 따라 달라질 수 있다. 플래시 메모리, 동급 비-휘발성 메모리 또는 광 디스 크 드라이브들 등과 같은 기타 내장 하드웨어 또는 주변 장치들이 도 1-2에 도시된 하드웨어 대신에 또는 추가 로 사용될 수 있다. 또한, 예시적 실시 예들의 프로세스들은 멀티프로세서 데이터 처리 시스템에 적용될 수도 있다.
- [0092] [0090] 일부 예시적 예들에서, 데이터 처리 시스템(200)은, 운영 체제 파일들 및/또는 사용자 생성 데이터를 저 장하기 위한 비-휘발성 메모리를 제공하기 위해 일반적으로 플래시 메모리로 구성되는, 개인 디지털 지원장치 (Personal Digital Assistant: PDA)일 수 있다. 버스 시스템은 시스템 버스, I/O 버스 및 PCI 버스와 같은, 하 나 또는 그 이상의 버스들로 구성될 수 있다. 물론, 상기 버스 시스템은 패브릭 또는 아키텍처에 부착된 서로 다른 컴포넌트들 또는 디바이스들 간의 데이터 전송을 제공하는 모든 유형의 통신 패브릭 또는 아키텍처를 사용 하여 구현될 수 있다.
- [0093] [0091] 통신 유닛은, 모뎀이나 네트워크 어댑터와 같은, 데이터 전송 및 수신에 사용되는 하나 또는 그 이상의 디바이스들을 포함할 수 있다. 메모리는, 예를 들어, 메인 메모리(208)이거나, 노스 브리지 및 메모리 컨트롤러 허브(202)에서 발견되는 캐시와 같은, 캐시일 수 있다. 처리 유닛은 하나 또는 그 이상의 프로세서들 또는 CPU 들을 포함할 수 있다.
- [0094] [0092] 도 1-2에 도시된 예들과 위에서 설명한 예들은 아키텍처의 제한들을 암시하려고 한 것이 아니다. 예를 들어, 데이터 처리 시스템(200)은 모바일 또는 웨어러블 디바이스 형태 외에 태블릿 컴퓨터, 노트북 컴퓨터 또 는 전화 디바이스일 수도 있다.
- [0095] [0093] 컴퓨터 또는 데이터 처리 시스템이 가상 시스템, 가상 디바이스 또는 가상 컴포넌트로 설명되는 경우, 상기 가상 시스템, 상기 가상 디바이스 또는 상기 가상 컴포넌트는 데이터 처리 시스템(200)에 설명된 일부 또 는 모든 컴포넌트들의 가상화 된 표현(manifestation)을 사용하여 데이터 처리 시스템(200)의 방식으로 동작하 는 것으로 설명된다. 예를 들어, 가상 시스템, 가상 디바이스 또는 가상 컴포넌트에서, 처리 유닛(206)은 호스 트 데이터 처리 시스템에서 이용 가능한 하드웨어 처리 유닛들(206)의 전부 또는 일부의 가상화 된 인스턴스로서 표현되고(manifested as a virtualized instance), 메인 메모리(208)은 호스트 데이터 처리 시스템에서 이 용 가능한 메인 메모리(208)의 전부 또는 일부의 가상화 된 인스턴스로서 표현되며, 그리고 디스크(226)는 호스 트 데이터 처리 시스템에서 이용할 수 있는 디스크(226) 전체 또는 일부의 가상화 된 인스턴스로서 표현된다. 그러한 경우들에서, 상기 호스트 데이터 처리 시스템은 데이터 처리 시스템(200)을 나타낸다.
- [0096] [0094] 도 3을 참조하면, 이 도면은 일 예시적인 실시예에 따른, 양자 컴퓨팅에서 어려운 변분 양자 문제들 (hard variational quantum problems)을 해결하기 위해 중간 재-최적화로 단열 진행(adiabatic progression)을

위한 예시적인 구성의 블록도를 묘사한다. 클라우드(300)은 도 1의 QCE (140)의 예이다. 고전적 컴퓨팅 환경(310)은 도 1의 CCN(104)의 예이다. 양자 컴퓨팅 환경(320)은 도 1의 QCN(146)의 예이다. 애플리케이션(312)은 도 1의 애플리케이션(105)의 예들이고 도 1의 서버(104), 또는 도 1의 임의의 다른 적절한 디바이스에서 실행된다.

- [0097] [0095] 고전적 컴퓨팅 환경(310) 내에서, 애플리케이션(312)은 입력으로서 단일 연속 파라미터  $z$ 를 갖는 파라미터화된 문제를 수신한다. 애플리케이션(312)은  $z$ 의 다양한 값에 대응하는 양자 회로를 생성한다.
- [0098] [0096] 그 다음, 양자 컴퓨팅 환경(320) 내에서, 큐비트들(344) 및 관독 회로(346)를 포함하는 양자 프로세서(340)는 양자 회로를 실행한다. 양자 컴퓨팅 환경(320)은 또한 양자 프로세서 지원 모듈(330)을 포함하고, 이것은 양자 회로를 실행한 결과들을 원래의 입력 문제 디스크립션(the original input problem description)에 대한 솔루션으로서 출력한다.
- [0099] [0097] 도 4를 참조하면, 이 도면은 일 예시적인 실시예에 따른, 양자 컴퓨팅에서 어려운 변분 양자 문제들을 해결하기 위해 중간 재-최적화로 단열 진행을 위한 예시적인 구성의 블록도를 도시한다. 애플리케이션(312)은 도 3의 애플리케이션 (312)과 동일하다.
- [0100] [0098] 애플리케이션(312)은 알려진 쉬운 문제를, 입력으로서, 수신한다. 쉬운 문제는 대응 연속 파라미터  $z$ 를 갖는다. 애플리케이션(312)은 또한 해결되어야 하는 문제의 원하는 복잡성을 나타내는  $z$ 의 목표 값을, 입력으로서, 수신한다.
- [0101] [0099] 단계 생성기(410)는 연속 파라미터  $z$ 에서 하나 또는 그 이상의 증가들을 생성한다.
- [0102] 리와인드 실시예를 구현할 때, 모듈(410)은 쉬운 상태에서 연속 파라미터  $z$ 로 시작하고,  $z$ 를 목표 상태(즉, 최대  $z$ )까지 증가시킨다. 만일  $z$ 의 증가가 너무 큰 것으로 판명되면, 모듈(410)은 시작 지점으로부터  $z$ 에서 더 작은 증가를 선택한다. 만일 새로운  $z$ 가 문제를 해결하는 데 성공하면, 모듈(410)은 새로운  $z$ 를 시작 지점으로 만들고,  $z$ 를 다시 한 번 목표 상태까지 또는 새로운 시작 지점 또는 목표 상태 사이의 다른 중간 상태까지 증가시킨다. 모듈(410)은 애플리케이션(312)이  $z$ 의 목표 값을 갖는 올바른 양자 회로 구성을 생성할 때까지 각 단계에서  $z$ 를 계속해서 증가시킨다.
- [0103] [0100] 레이스 실시예를 구현할 때, 모듈(410)은 쉬운 상태의 연속 파라미터  $z$ 로 시작하고, 시작 지점으로부터  $z$ 에서 한 세트의 증가들 생성한다. 일단 대응하는 양자 회로들의 그룹이 실행을 완료하면, 모듈(410)은 솔루션들 그룹에서 하나의 회로에 대응하는  $z$ 의 가장 큰 값을 선택한다. 모듈(410)은 상기 새로운  $z$ 를 시작 지점으로 만들고,  $z$ 에서 새로운 증가들 세트를 생성한다. 모듈(410)은 애플리케이션(312)이  $z$ 의 목표 값을 갖는 올바른 양자 회로 구성을 생성할 때까지 각 단계에서  $z$ 를 계속해서 증가시킨다.
- [0104] [0101] 파라미터 세트 생성기(420)는 단계 생성기(410)에 의해 생성된  $z$ 의 각각의 증가에 대응하는 한 세트의 변분 파라미터들을 생성한다. 상기 변분 파라미터들의 세트는 문제를 해결하기 위해 사용될 수 있는 양자 회로의 구성을 정의한다.
- [0105] [0102] 문제 해결사 인터페이스 모듈(430)은, 상기 변분 파라미터들 세트에 의해 정의된 양자 회로 구성을, 입력으로 취한다. 모듈(430)은 실행을 위해 상기 양자 회로 구성을 양자 컴퓨팅 환경(320)에 전달하고, 양자 컴퓨팅 환경(320)으로부터 실행 결과들을 다시 수신한다. 모듈(430)은 또한 양자 회로 구성에 대한 실행 시간 및 완료 시간을 모니터링한다.
- [0106] [0103] 솔루션 검증기 인터페이스 모듈(440)은 양자 회로의 출력을 다른 수단(예를 들어, 솔루션의 다항-시간 검증의 일부 형태를 사용하거나, 정확한 고전 솔루션과 비교함으로써)에 의해 획득된 문제에 대해 알려진 올바른 솔루션과 비교한다.
- [0107] [0104] 도 5를 참조하면, 이 도면은 예시적인 실시예에 따라, 양자 컴퓨팅에서 어려운 변분 양자 문제를 해결하기 위해 중간 재-최적화를 사용하는 단열 진행을 사용하여 해결하기에 적합한 어려운 변분 양자 문제의 예를 도시한다.
- [0108] [0105] 그래프(510)는 불소 분자에 대한 해리 곡선을 도시한다. 각 불소 분자는 두 개의 불소 원자들로 구성된다. 그래프(510)의 x-축은 두 원자들 사이의 거리이고, 그래프(510)의 y-축은 에너지의 양이다. 따라서, 그래프(510)는 원자들이 서로 특정 거리에 있을 때 두 원자들을 떼어내는 데 필요한 에너지의 양을 나타낸다(plot). 해리 곡선에서 각 지점을 계산하는 것은 특정 복잡성의 변분 문제이다.

- [0109] [0106] 그래프(520)는 그래프(510)에 대응하고, 총 스핀 및 총 에너지의 2가지 오류 측정치들을 도시하며, 이는 종래의 방법들을 사용하여 그래프(510)를 계산하는 것과 연관된, 고전적 계산과 비교된다. 특히, 영역(530)은 해리 곡선 상의 계산 지점들에서의 오류들이 기준선 위로 스파이크하는 영역을 나타낸다. 따라서, 영역(530)의 왼쪽에 있는 해리 곡선의 x-축 상의 지점은 풀기가 더 쉬운 문제(an easier problem to solve)에 대응한다. 영역(530)의 오른쪽에 있는 해리 곡선의 x-축 상의 지점은 비교적 풀기 어려운 문제(a relatively harder problem to solve)에 대응한다. 영역(530) 내의 해리 곡선의 x-축 상의 지점은 가장 풀기 어려운 문제에 대응한다. 만일 해리 곡선을 계산하는 문제가 변분 문제로 파라미터화되면, x-축 상에서 영역(530)의 왼쪽 지점들은 z의 낮은 값을 갖고, 영역(530)의 오른쪽 지점들은 중간-척도 값의 z를 가지며, 영역(530) 내의 지점들은 높은 값의 z를 갖는데, 여기서 z 값들은 공통 척도 범위에 있고, 예를 들어 0-1에 있다. 불소에 대한 해리 곡선은 비-제한적인 예이다. 다른 많은 해리 곡선들이 유사한 특성들을 가지며, 그러한 곡선들의 지점들을 구하는 것은 유사한 양의 비교 계산 복잡성을 제공한다.
- [0110] [0107] 도 6을 참조하면, 이 도면은 예시적인 실시예에 따른, 중간 재-최적화를 사용하는 단일 진행을 사용하여 최대 컷 문제(a max cut problem)를 해결하는 예를 도시한다. 상기 문제는 여기에 설명된 방식으로, 도 3의 애플리케이션(312)을 사용하여 해결될 수 있다.
- [0111] [0108] 도시된 바와 같은, 각각의 반복은 z 값에 대한 위치를 나타내고, z 값은 맥스컷과 같은 그래프 상의 최적화 문제를 파라미터화하는 연속 파라미터로서, 0에서 1까지의 값이다. z = 0일 때 그래프는 한 개의 엣지를 갖고(문제는 쉽게 풀 수 있음), z = 1일 때 그래프는 원래 문제 인스턴스의 모든 엣지들(the original problem instance)을 갖는다.
- [0112] [0109] 반복(600)에서, 애플리케이션(312)은, 리와인드 방법을 사용하여, 쉬운 상태(0)에서, 연속 파라미터 z로 시작하여, z를 목표 상태(즉, 최대 z 또는 1)로 증가시키는 단계 0을 수행한다. 그런 다음 상기 애플리케이션은 한 세트의 변분 파라미터들에 의해서 정의된 양자 회로 구성을 사용하여 문제를 해결하려고 시도한다. 상기 애플리케이션은 양자 회로의 출력을 다른 수단들에 의해서 획득된 문제에 대해 알려진 올바른 솔루션과 비교하고, 단계 0의 오류가  $5.16 * 10^{-7}$ 로서, 알려진-올바른 솔루션과 미리 정의된 임계 차이값 보다 높다고 결정한다. 따라서, 상기 애플리케이션은, z를 0에서 1로 증가시키는, 단계 0은, 너무 컷다고 결론을 내린다.
- [0113] [0110] 반복(610)에서, 상기 애플리케이션은 단계1을 수행하여, 시작 지점으로부터 더 작은 증가를 z에서 선택하는데, 여기서는 이전 증가 크기의 절반 또는 0.5를 선택한다. 그런 다음 상기 애플리케이션은 한 세트의 변분 파라미터들에 의해서 정의된 양자 회로 구성을 사용하여 문제를 해결하려고 시도한다. 상기 애플리케이션은 양자 회로의 출력을 다른 수단들에 의해서 획득된 문제에 대해 알려진 올바른 솔루션과 비교하고, 단계1의 오류가,  $3.42 * 10^{-8}$ 로서, 알려진-올바른 솔루션과 미리 정의된 임계 차이값보다 높다고 결정한다. 따라서, 상기 애플리케이션은, z를 0에서 0.5로 증가시키는, 단계1도, 또한 너무 컷다고 결론을 내린다.
- [0114] [0111] 반복(620)에서, 상기 애플리케이션은 단계2를 수행하여, 시작지점으로부터 더 작은 증가를 z에서 선택하는데, 여기서는 이전 증가 크기의 절반 또는 0.25를 선택한다. 그런 다음 상기 애플리케이션은 한 세트의 변분 파라미터들에 의해서 정의된 양자 회로 구성을 사용하여 문제를 해결하려고 시도한다. 이번에는, 단계2의 오류가  $4.56 * 10^{-9}$ 로서, 알려진-올바른 솔루션과 미리 정의된 임계 차이값보다 낮다.
- [0115] [0112] 그 결과, 반복(630)에서, 상기 애플리케이션은 단계3을 수행하여, z를 0.25에서 0.5로 증가시키고 대응 양자 회로 구성의 결과를 획득한다. 이번에도, 단계3의 오류가  $3.59 * 10^{-9}$ 로서, 알려진-올바른 솔루션과 미리 정의된 임계 차이값보다 여전히 낮다.
- [0116] [0113] 그 결과, 반복(640)에서, 상기 애플리케이션은 단계4를 수행하여, z를 0.5에서 1로 증가시키고, 대응 양자 회로 구성의 결과를 획득한다. 이번에도, 단계4의 오류가  $2.23 * 10^{-9}$ 로서, 또한 알려진-올바른 솔루션과 미리 정의된 임계 차이값보다 낮다. 오차가 임계 차이값보다 낮고 z = 1이므로, 문제의 목표 상태가 해결되었다.
- [0117] [0114] 도 6에 도시된 증가들은 상기 애플리케이션이 목표 상태에 도달하기 위해 수행할 수 있었던 추가의 단계들을 도시하지는 않았다. 상기 애플리케이션은 예시적인 실시예들의 범위를 벗어나지 않고 추가의 단계들, 더 적은 단계들, 또는 다른 크기들의 증가들을 수행할 수 있다.
- [0118] [0115] 도 7을 참조하면, 이 도면은 예시적인 실시예에 따라, 중간 재-최적화를 사용하는 단일 진행을 사용하여

해리 곡선을 계산하는 예를 묘사한다. 여기에 설명된 방식으로, 곡선 상의 지점들은 도 3의 애플리케이션(312)을 사용하여 해결될 수 있다.

- [0119] [0116] 그래프(700)는 애플리케이션(312)을 사용하여 얻은 해리 곡선 상의 지점들을 묘사한다. 그래프(700)의 x-축은 앙스트롬들(Angstroms)로 측정된 원자간 거리를 나타내고, y-축은 해리 에너지를 나타낸다. 그래프(700)는 또한 다른 방법을 사용하여 획득된, 이상적인 해리 곡선(710)을 도시한다.
- [0120] [0117] 영역(720)은 높은 값의 z(예: 0-1 척도에서 1에 가까움)로 파라미터화된, 더 높은 복잡성을 갖는 것으로 알려진 x-축의 일부분을 도시한다. 그 결과, 상기 애플리케이션은 곡선 상의 지점들에 대해 해결하기(solving)를 시작하는데 x-축 상의 왼쪽으로 더(further) 시작한다.
- [0121] [0118] 구체적으로, 상기 애플리케이션은 지점(730)에서 시작하는데, 이 지점은 해결되었다. 단계(740)에서, 상기 애플리케이션은, 리와인드 방법을 사용하여, z를 증가시켜 지점(730)의 x 값보다 더 높은 x값을 갖는 지점에 대해 해결 하려고 시도한다. 그러나, 상기 애플리케이션의 초기 솔루션은 동일한 x 값을 갖는 곡선(710) 상의 지점으로부터 임계 차이값보다 높다.
- [0122] [0119] 그 결과, 단계(750)에서 상기 애플리케이션은 단계(740)의 증가보다 더 작은 증가를 z에서 생성한다. 더 작은 증가는 동일한 x 값을 갖는 곡선(710) 상의 지점으로부터 임계 차이값보다 낮은 솔루션을 초래한다.
- [0123] [0120] 그 결과, 단계(760)의 z값이 새로운 출발점이 된다. 단계(760)에서 상기 애플리케이션은 단계(740)에서 시도된 것과 동일한 레벨로, z를 다시 한 번 증가시킨다. 단계(750)는 또한 동일한 x 값을 갖는 곡선(710) 상의 지점으로부터 임계 차이값 보다 낮은 솔루션을 초래한다. 상기 애플리케이션은 해리 곡선의 모든 원하는 지점들이 해결될 때까지 여기에 설명된 방식으로 계속한다.
- [0124] [0121] 도 8을 참조하면, 이 도면은 예시적인 실시예에 따른, 중간 재-최적화를 갖는 단일 진행을 사용하여 해리 곡선을 계산하는 다른 예를 도시한다. 곡선 상의 지점들은 여기에 설명된 방식으로 도 3의 애플리케이션(312)을 사용하여 해결될 수 있다. 그래프(800), 곡선(710), 및 지점(730)은 도 7의 그래프(700), 곡선(710) 및 지점(730)과 동일하다.
- [0125] [0122] 상기 애플리케이션은, 해결된 지점(730)에서 시작된다. 상기 애플리케이션은, 레이스 방법을 사용하여, 한 세트의 증가들을 z에서 생성하여 지점(730)의 값보다 높은 x 값들을 갖는 지점들에 대해 해결하기를 시도한다. 상기 애플리케이션은 z에서 증가들 세트에 대응하는 양자 회로들을 생성하고, 양자 회로들이 실행을 완료하는 미리 결정된 시간 기간 동안 대기한다. 상기 시간 기간이 종료될 때, 영역(840)에 대응하는 모든 양자 회로들은 실행을 완료한다. x-축을 따라 더 오른쪽에 있는 지점들, 예를 들어, 지점들(860 및 870)에 대응하는 양자 회로들은 실행을 완료하지 않았다. 지점(850)은 영역(840) 내의 하나의 회로에 대응하는 지점(730)으로부터 z에서 가장 높은 증가를 나타내기 때문에, 상기 애플리케이션은 지점(850)에 대응하는 z 값을 새로운 시작지점으로 선택하고 프로세스를 반복한다. 상기 애플리케이션은 해리 곡선 상의 모든 원하는 지점들이 해결될 때까지, 여기에 설명된 방식으로, 계속한다.
- [0126] [0123] 도 9를 참조하면, 이 도면은 예시적인 실시예에 따른 양자 컴퓨팅에서 어려운 변분 양자 문제들을 해결하기 위해 중간 재-최적화를 사용하는 단일 진행을 위한 예시적인 프로세스의 플로를 도시한다. 프로세스(900)는 도 3 또는 도 4의 애플리케이션(312)에서 구현될 수 있고, 여기에 설명된 리와인드 방법의 연산을 예시한다.
- [0127] [0124] 블록(902)에서, 상기 애플리케이션은 파라미터화된 최적화 문제의 연속 파라미터에서 증가를 생성하며, 상기 연속 파라미터에서 현재 상태는 문제의 해결 가능한 상태에 대응한다. 블록(904)에서, 상기 애플리케이션은 상기 증가에 대응하는 한 세트의 변분 파라미터들을 생성한다. 블록(906)에서, 상기 애플리케이션은 문제를 해결하기 위해 상기 변분 파라미터들 세트에 의해 명시된 양자 회로 구성을 사용한다. 블록(908)에서, 상기 애플리케이션은 문제의 솔루션이 문제에 대해 알려진 올바른 솔루션과 다른 임계값보다 큰지를 체크한다. 만일 그렇지 않다면(블록 908의 "아니오" 경로), 블록(910)에서, 상기 애플리케이션은, 문제의 연속 파라미터에서, 이전 증가보다 작은 다른 증가를 생성한 다음, 블록(904)로 돌아간다. 만일 그렇다면(블록 908의 "예" 경로), 블록(912)에서 상기 애플리케이션은 연속 파라미터 z가 목표 상태에 있는지를 체크한다. 만일 "예"라면, 상기 애플리케이션은 종료된다. 만일 그렇지 않으면(블록 912의 "아니오" 경로), 상기 애플리케이션은 블록(902)로 돌아간다.
- [0128] [0125] 도 10을 참조하면, 이 도면은 예시적인 실시예에 따른 양자 컴퓨팅에서 어려운 변분 양자 문제들을 해결하기 위해 중간 재-최적화를 사용하는 단일 진행을 위한 다른 예시적인 프로세스의 플로를 도시한다. 프로세스(1000)는 도 3 또는 도 4의 애플리케이션(312)에서 구현될 수 있고, 여기에 설명된 레이스 방법의 동작을 예시

한다.

- [0129] [0126] 블록(1002)에서, 상기 애플리케이션은 파라미터화된 최적화 문제의 연속 파라미터에서 한 세트의 다른-크기의 증가들을 생성하며, 연속 파라미터에서 현재 상태는 문제의 해결 가능한 상태에 대응한다. 블록(1004)에서, 상기 애플리케이션은 각각의 증가에 대응하는 한 세트의 변분 파라미터들을 생성한다. 블록(1006)에서, 상기 애플리케이션은 문제를 해결하기 위해 변분 파라미터들 세트의 각각에 의해 명시된 양자 회로 구성을 사용한다. 블록(1008)에서, 상기 애플리케이션은 양자 회로들 중 하나가 실행을 완료한 후 미리 결정된 시간 기간 동안 대기한다. 블록(1010)에서, 상기 애플리케이션은 상기 시간 기간 내에 실행을 완료한 연속 파라미터에서 가장 큰 증가에 대응하는 양자 회로를 선택하고  $z$ 의 현재 값을 상기 가장 큰 증가로 대체한다. 블록(1012)에서, 상기 애플리케이션은 연속 파라미터  $z$ 가 목표 상태에 있는지를 체크한다. 만일 "예"라면, 상기 애플리케이션은 종료된다. 만일 그렇지 않으면(블록 1012의 "아니오" 경로), 상기 애플리케이션은 블록(1002)로 돌아간다.
- [0130] [00127] 따라서, 컴퓨터 구현 방법, 시스템, 또는 장치, 및 컴퓨터 프로그램 제품이 온라인 공동체들 및 기타 관련 특징들, 기능들, 또는 연산들의 참여를 관리하기 위한 예시 적 실시 예들에 제공된다. 하나의 실시 예 또는 그 일부가 디바이스의 유형과 관련하여 기술되는 경우, 컴퓨터 구현 방법, 시스템, 또는 장치, 컴퓨터 프로그램 제품, 또는 그 일부는 디바이스의 해당 유형에 대한 적합하고 비교 가능한 표현으로 사용을 위해 구체화되거나 구성된다.
- [0131] [00128] 실시 예가 애플리케이션으로 기술되는 경우, 소프트웨어 서비스(Software as a Service)(SaaS)로 상기 애플리케이션의 전달이 예시적 실시 예들의 범위 내에서 고려된다. SaaS 모델에서, 실시 예를 구현하는 애플리케이션의 능력이 클라우드 인프라스트럭처에서 상기 애플리케이션을 실행함으로써 사용자에게 제공된다. 사용자는 웹브라우저(예를들어, 웹-기반 이-메일)와 같은 썬 클라이언트 인터페이스 또는 기타 경량의 클라이언트-애플리케이션들을 통한 다양한 클라이언트 디바이스들을 사용하여 상기 애플리케이션에 액세스할 수 있다. 사용자는 네트워크, 서버들, 운영체제들, 또는 클라우드 인프라스트럭처의 스토리지를 포함하는 하부 클라우드 인프라스트럭처를 관리하거나 컨트롤하지 않는다. 일부 경우들에서, 사용자는 SaaS 애플리케이션의 능력도 관리하거나 컨트롤하지 못할 수 있다. 일부 다른 경우들에서, 상기 애플리케이션의 SaaS 구현은 제한된 사용자-특화 애플리케이션 구성 세팅들의 가능한 예외를 허용할 있다.
- [0132] [00129] 본 발명은 시스템, 방법, 및/또는 통합의 모든 가능한 기술적 세부 레벨에서 컴퓨터 프로그램 제품일 수 있다. 컴퓨터 프로그램 제품은 프로세서로 하여금 본 발명의 실시 예들을 수행하게 하기 위한 컴퓨터 판독 가능 프로그램 명령을 갖는 컴퓨터 판독가능 저장 매체(또는 매체)를 포함할 수 있다.
- [0133] [00130] 상기 컴퓨터 판독 가능 스토리지 매체는 명령 실행 장치에 의해 사용될 명령들을 유지 및 저장할 수 있는 유형의(tangible) 디바이스일 수 있다. 상기 컴퓨터 판독 가능 스토리지 매체는, 예를 들면, 전자 스토리지 디바이스, 자기 스토리지 디바이스, 광 스토리지 디바이스, 전자기 스토리지 디바이스, 반도체 스토리지 디바이스, 또는 전송할 것들의 모든 적절한 조합일 수 있으며, 그러나 이에 한정되지는 않는다. 컴퓨터 판독 가능 스토리지 매체의 더 구체적인 예들의 비포괄적인 목록에는 다음이 포함될 수 있다: 휴대용 컴퓨터 디스켓, 하드 디스크, 랜덤 액세스 메모리(RAM), 판독-전용 메모리(ROM), 소거 및 프로그램가능 판독-전용 메모리(EPROM 또는 플래시 메모리), 정적 랜덤 액세스 메모리(SRAM), 휴대용 콤팩트 디스크 판독-전용 메모리(CD-ROM), 디지털 다용도 디스크(DVD), 메모리 스틱, 플로피 디스크, 천공-카드들 또는 명령들이 히스토리된 홈에 있는 용기된 구조들 같이 머신적으로 인코드 된 장치, 및 전송할 것들의 모든 적절한 조합. 본 명세서에서 사용될 때, 컴퓨터 판독 가능 스토리지 매체는 무선 전파들이나 다른 자유롭게 전파되는 전자기파들, 도파관이나 기타 전송 매체(예를 들어, 광섬유 케이블을 통해 전달되는 광 펄스들)를 통해 전파되는 전자기파들, 또는 선(wire)을 통해 전송되는 전기 신호들 같이 그 자체로 일시적인(transitory) 신호들로 해석되지는 않는다.
- [0134] [00131] 본 명세서에 기술되는 컴퓨터 판독 가능 명령들은, 예를 들어, 인터넷, 근거리 통신망, 광역 통신망 및/또는 무선 네트워크 등의 통신망(네트워크)을 통해 컴퓨터 판독 가능 스토리지 매체로부터 각각 컴퓨팅/처리 디바이스들로 또는 외부 스토리지 디바이스로부터 외부 컴퓨터로 다운로드 될 수 있다. 상기 통신망은 구리 전송 케이블들, 광 전송 섬유들, 무선 전송, 라우터들, 방화벽들, 스위치들, 게이트웨이 컴퓨터들 및/또는 엣지 서버들을 포함할 수 있다. 각 컴퓨팅/처리 장치 내 네트워크 어댑터 카드 또는 네트워크 인터페이스는 상기 통신망으로부터 컴퓨터 판독 가능 프로그램 명령들을 수신하고 그 컴퓨터 판독 가능 프로그램 명령들을 각각의 컴퓨팅/처리 디바이스 내의 컴퓨터 판독 가능 스토리지 매체에 저장하기 위해 전송한다.
- [0135] [00132] 본 발명의 연산들을 실행하기 위한 컴퓨터 판독 가능 프로그램 명령들은 Smalltalk, C++ 또는 그와 유사 언어 등의 객체 지향 프로그래밍 언어와 "C" 프로그래밍 언어 또는 그와 유사한 언어 등의 종래의 절차적 프

로그래밍 언어들을 포함하여, 하나 또는 그 이상의 프로그래밍 언어들을 조합하여 작성된(written) 어셈블러 명령들, 명령-세트-아키텍처(ISA) 명령들, 머신 명령들, 머신 종속 명령들, 마이크로코드, 펌웨어 명령들, 상태-셋팅 데이터, 집적회로를 위한 구성 데이터, 또는 소스 코드나 목적 코드일 수 있다. 상기 컴퓨터 판독 가능 프로그램 명령들은 전적으로 사용자의 컴퓨터상에서, 부분적으로 사용자의 컴퓨터상에서, 독립형(stand-alone) 소프트웨어 패키지로서, 부분적으로 사용자의 컴퓨터상에서 그리고 부분적으로 원격 컴퓨터상에서 또는 전적으로 원격 컴퓨터나 서버상에서 실행될 수 있다. 위에서 마지막의 경우에, 원격 컴퓨터는 근거리 통신망(LAN) 또는 광역 통신망(WAN)을 포함한 모든 종류의 네트워크를 통해서 사용자의 컴퓨터에 접속될 수 있고, 또는 이 접속은 (예를 들어, 인터넷 서비스 제공자를 이용한 인터넷을 통해서) 외부 컴퓨터에 이루어질 수도 있다. 일부 실시예들에서, 예를 들어 프로그램 가능 논리 회로, 필드-프로그램 가능 게이트 어레이들(FPGA), 또는 프로그램 가능 논리 어레이들(PLA)을 포함한 전자 회로는 본 발명의 실시 예들을 수행하기 위해 전자 회로를 맞춤화하도록 상기 컴퓨터 판독 가능 프로그램 명령들의 상태 정보를 활용하여 상기 컴퓨터 판독 가능 프로그램 명령들을 실행할 수 있다.

[0136] [00133] 본 명세서에서는 본 발명의 실시 예들에 따른 방법들, 장치들(시스템들), 및 컴퓨터 프로그램 제품들의 플로 차트 예시도들 및/또는 블록 다이어그램들을 참조하여 본 발명의 실시 예들을 기술한다. 플로 차트 예시도들 및/또는 블록 다이어그램들의 각 블록과 플로 차트 예시도들 및/또는 블록 다이어그램들 내 블록들의 조합들은 컴퓨터 판독 가능 프로그램 명령들에 의해 구현될 수 있다는 것을 이해할 수 있을 것이다.

[0137] [00134] 이들 컴퓨터 판독 가능 프로그램 명령들은 범용 컴퓨터, 특수목적용 컴퓨터, 또는 기타 프로그램가능 데이터 처리 장치의 프로세서에 제공되어 머신(machine)을 생성하고, 그렇게 하여 그 명령들이 상기 컴퓨터 또는 기타 프로그램가능 데이터 처리 장치의 프로세서를 통해서 실행되어, 상기 플로 차트 및/또는 블록 다이어그램의 블록 또는 블록들에 명시된 기능들/동작들을 구현하기 위한 수단을 생성할 수 있다. 이들 컴퓨터 판독 가능 프로그램 명령들은 또한 컴퓨터 판독 가능 스토리지 매체에 저장될 수 있으며, 컴퓨터, 프로그램가능 데이터 처리 장치 및/또는 기타 디바이스들에 지시하여 명령들이 저장된 상기 컴퓨터 판독 가능 스토리지 매체가 상기 플로 차트 및/또는 블록 다이어그램의 블록 또는 블록들에 명시된 기능/동작의 특징들을 구현하는 명령들을 포함하는 제조품(an article of manufacture)을 포함하도록 특정한 방식으로 기능하게 할 수 있다.

[0138] [00135] 상기 컴퓨터 판독 가능 프로그램 명령들은 또한 컴퓨터, 기타 프로그램가능 데이터 처리 장치, 또는 다른 디바이스에 로드 되어, 상기 컴퓨터, 기타 프로그램가능 장치 또는 다른 디바이스에서 한 세트의 동작 단계들이 수행되게 하여 컴퓨터 구현 프로세스를 생성하며, 그렇게 하여 상기 컴퓨터, 기타 프로그램가능 장치, 또는 다른 디바이스 상에서 실행되는 명령들이 플로 차트 및/또는 블록 다이어그램의 블록 또는 블록들에 명시된 기능들/동작들을 구현할 수 있다.

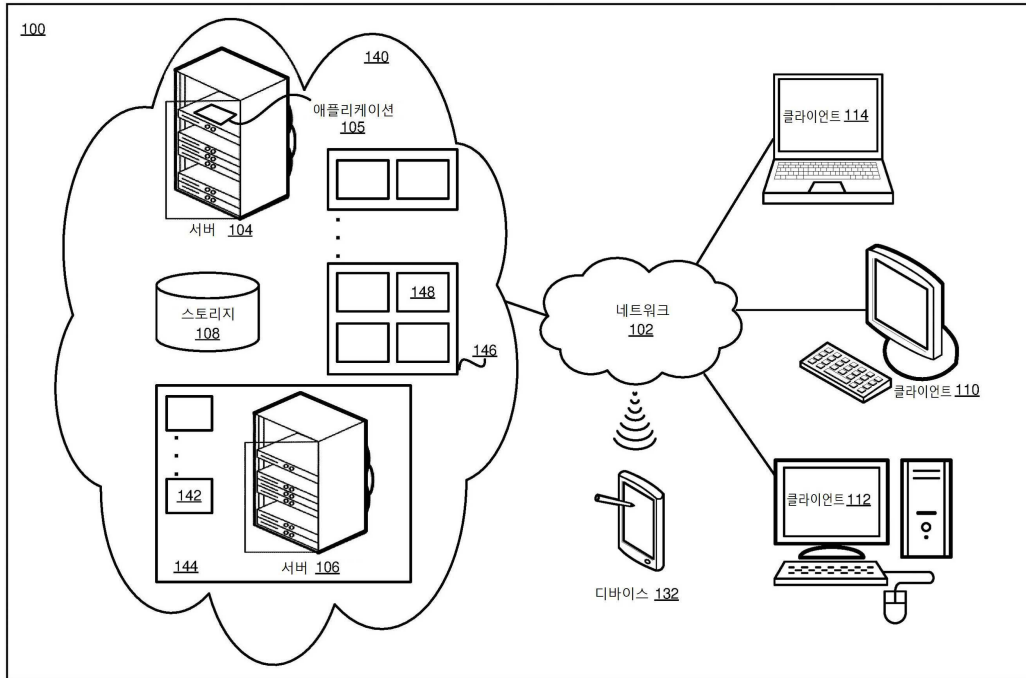
[0139] [00136] 도면들 내 플로 차트 및 블록 다이어그램들은 본 발명의 여러 실시 예들에 따른 시스템들, 방법들 및 컴퓨터 프로그램 제품들의 가능한 구현들의 아키텍처, 기능(functionality), 및 연산(operation)을 예시한다. 이와 관련하여, 상기 플로 차트 또는 블록 다이어그램들 내 각 블록은 상기 명시된 논리적 기능(들)을 구현하기 위한 하나 또는 그 이상의 실행 가능한 명령들을 포함한 모듈, 세그먼트 또는 명령들의 일부분을 나타낼 수 있다. 일부 다른 구현들에서, 상기 블록에 언급되는 기능들은 도면들에 언급된 순서와 다르게 일어날 수도 있다. 예를 들면, 연속으로 도시된 두 개의 블록들은 실제로는 사실상 동시에 실행될 수도 있고, 또는 이 두 블록들은 때때로 연관된 기능에 따라서는 역순으로 실행될 수도 있다. 블록 다이어그램들 및/또는 순서 예시도의 각 블록, 및 블록 다이어그램들 및/또는 순서 예시도 내 블록들의 조합들은 특수목적용 하드웨어 및 컴퓨터 명령들의 명시된 기능들 또는 동작들, 또는 이들의 조합들을 수행하는 특수목적용 하드웨어-기반 시스템들에 의해 구현될 수 있다는 것에 또한 주목해야 한다.

[0140] [0137] 본 발명의 실시 예들은 또한 고객 기업, 비영리 단체, 정부기관, 내부 조직 구조, 또는 이와 유사한 조직과 맺은 서비스 계약의 일부로서 전달될 수 있다. 이들 실시 예들의 특징들은 수행할 컴퓨터 시스템의 구성, 소프트웨어, 하드웨어의 배치, 그리고 여기서 기술한 방법들의 일부, 또는 전부를 구현하는 웹 서비스들을 포함한다. 이들 실시 예들의 특징들은 또한 고객의 운영들을 분석하는 것, 상기 분석에 응답하여 추천들을 생성하는 것, 상기 추천들의 부분들을 구현하는 시스템들을 구축하는 것, 시스템들을 기존의 프로세스들 및 인프라스트럭처에 통합하는 것, 상기 시스템들의 사용을 계측하는 것, 상기 시스템의 사용자들에게 비용들을 할당하는 것, 및 상기 시스템들의 사용에 대한 청구서를 발행하는 것을 포함한다. 본 발명의 상기 실시 예들 각각은 그들 개별 장점들을 진술함으로써 각각 기술되었지만, 본 발명이 그들에 관한 특정 조합에 국한되는 것은 아니다. 이와 달리, 그러한 실시 예들은 또한 그들의 유익한 효과들을 상실함이 없이 본 발명의 의도된 배치에 따라 임의의

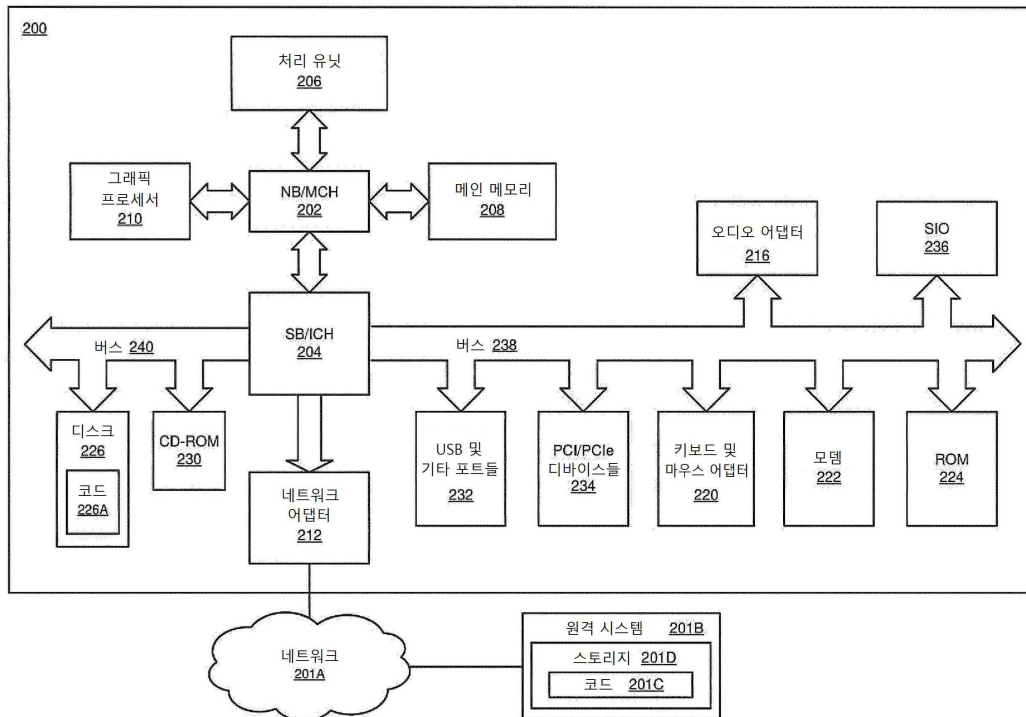
방법 및 수로 조합될 수 있다.

도면

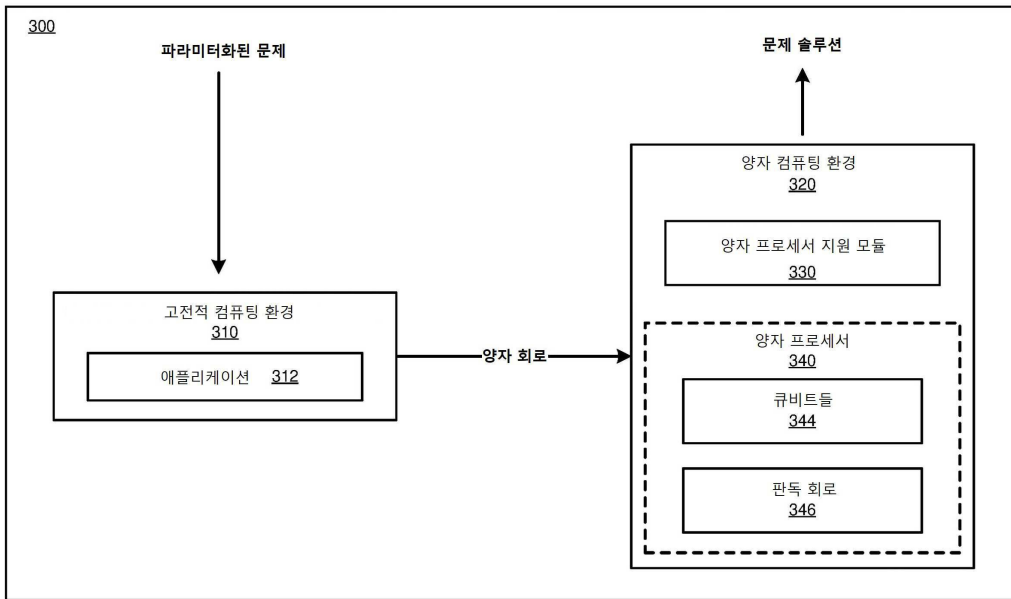
도면1



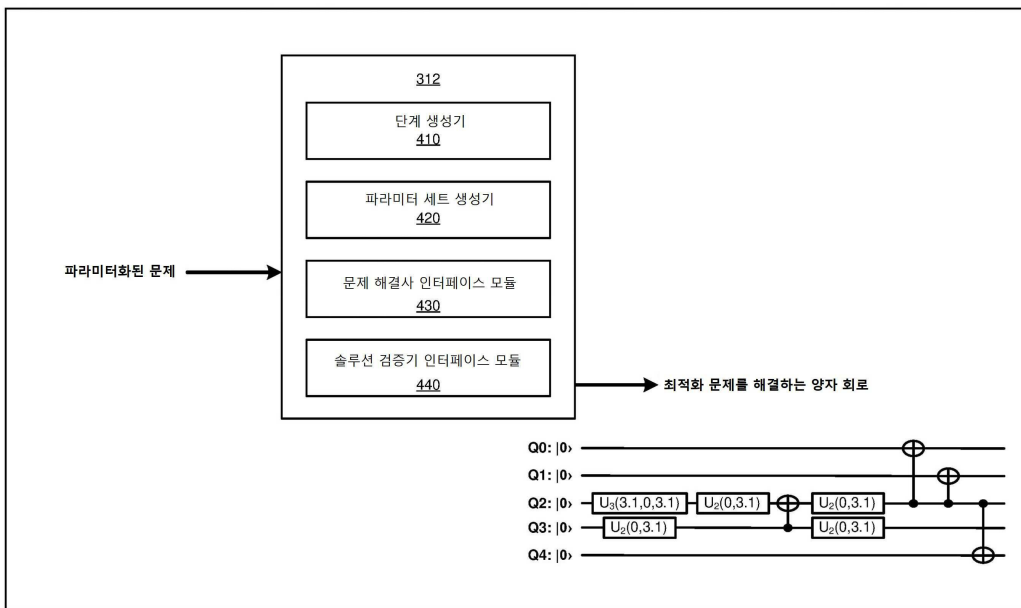
도면2



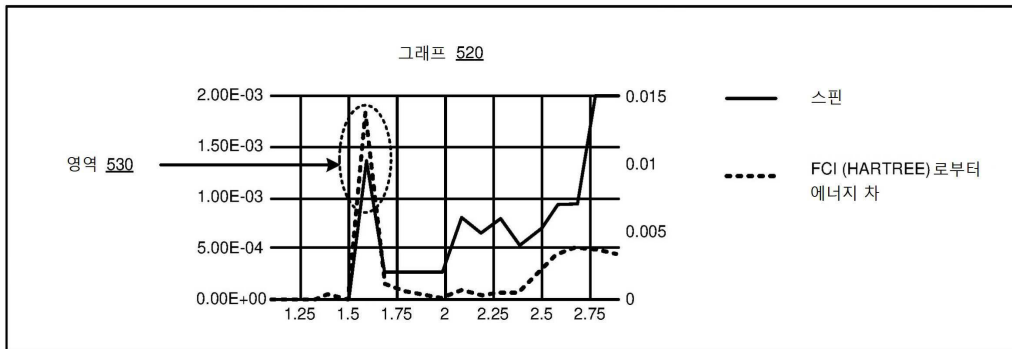
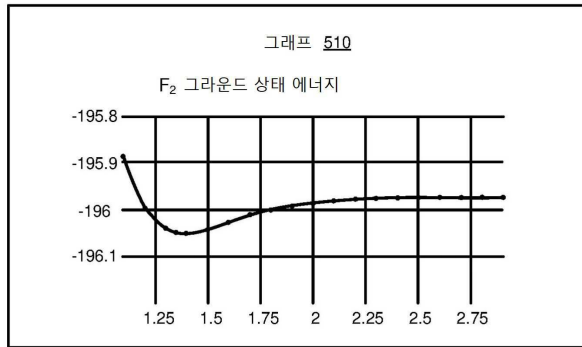
도면3



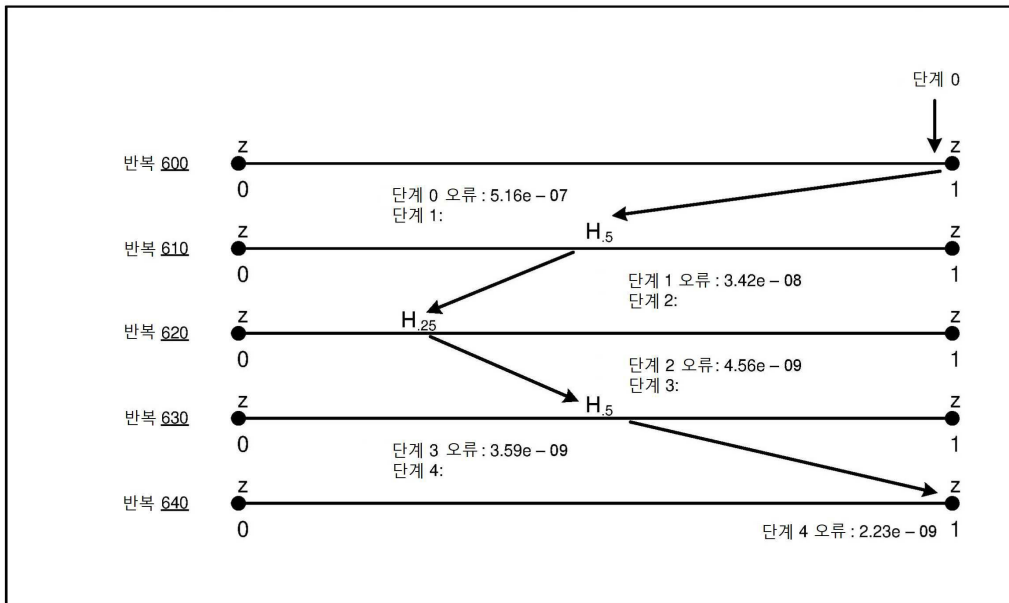
도면4



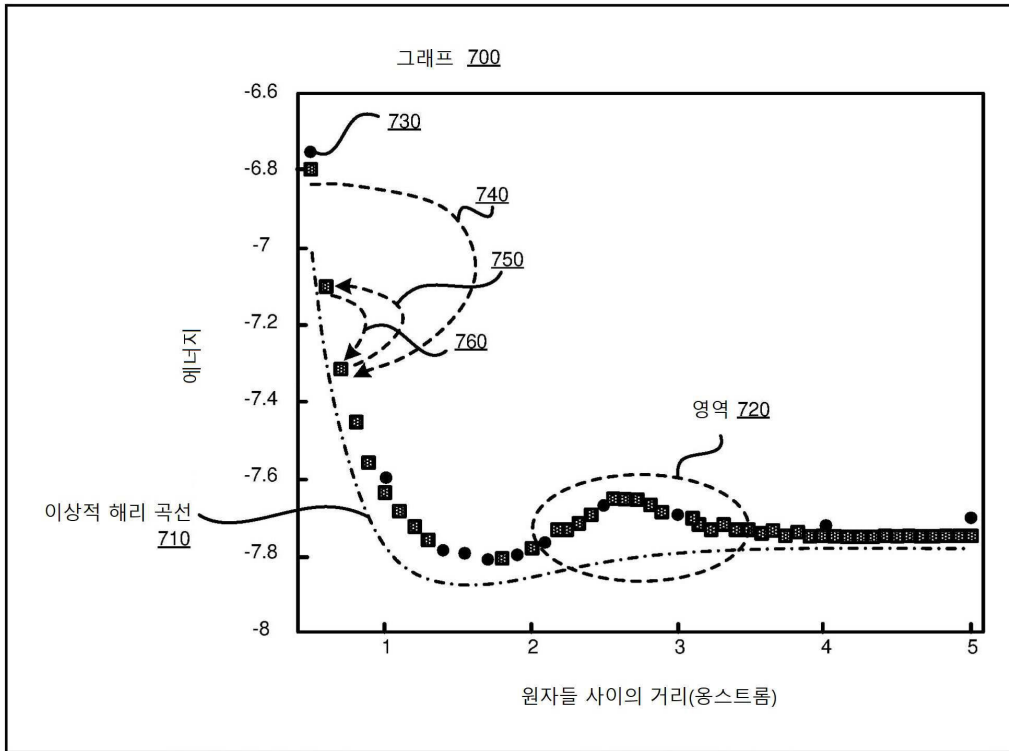
도면5



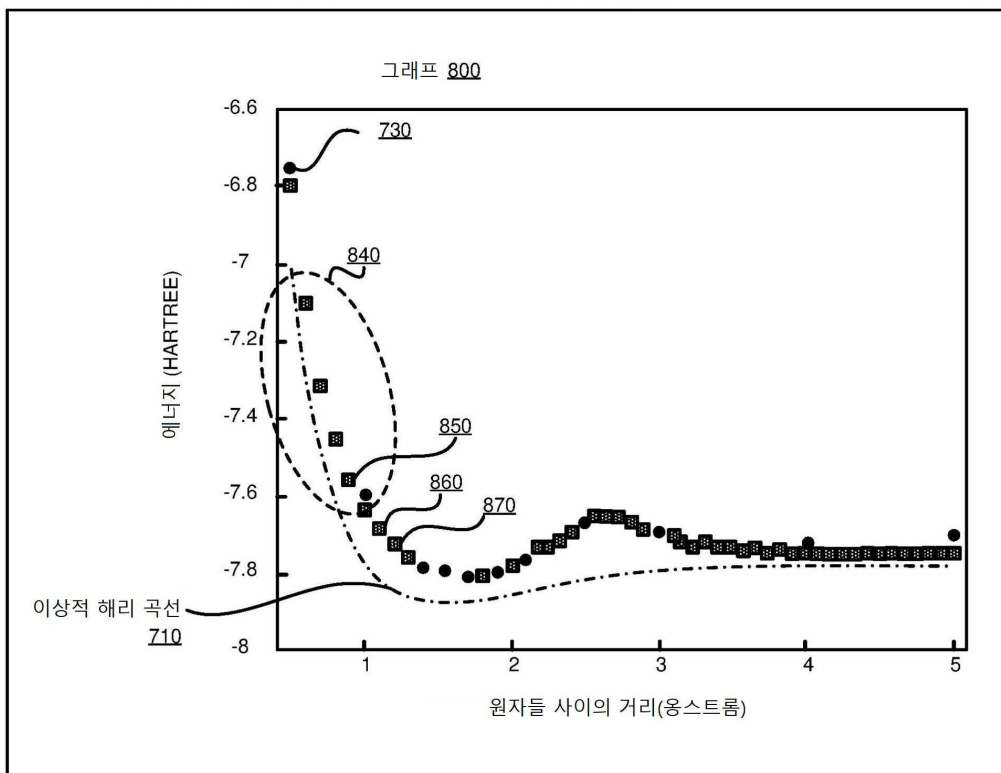
도면6



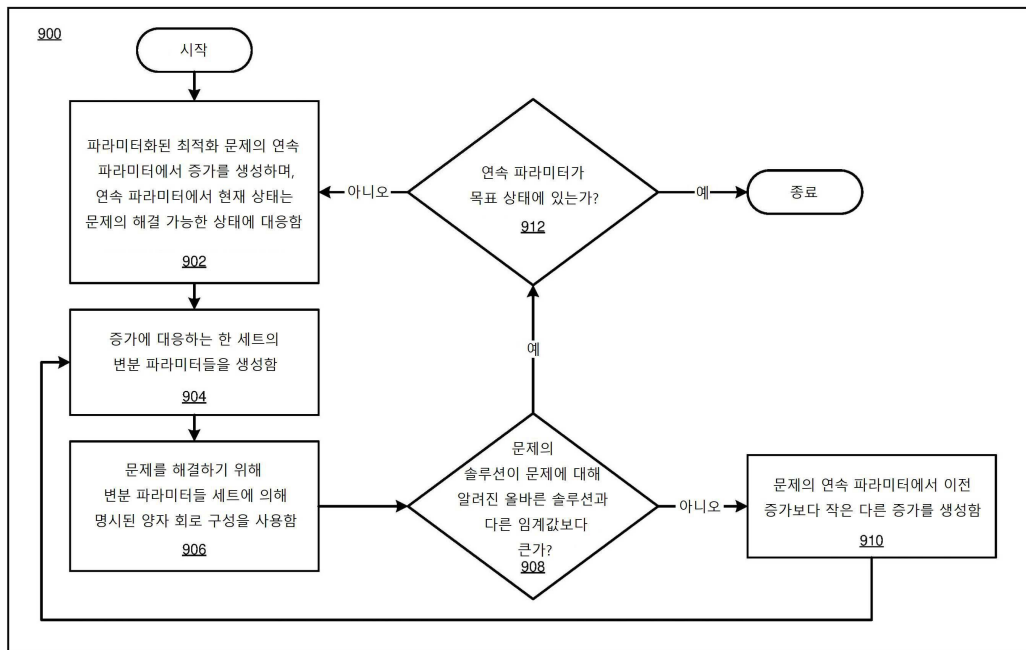
도면7



도면8



도면9



도면10

