(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
29 August 2024 (29.08.2024)

WIPO | PCT

(10) **International Publication Number**
WO 2024/178046 A1

(51) **International Patent Classification:**
G01V 1/28 (2006.01) ***G06N 10/00*** (2022.01)
G06F 30/20 (2020.01) *G01V 20/00* (2024.01)

(21) **International Application Number:**
PCT/US2024/016621

(22) **International Filing Date:**
21 February 2024 (21.02.2024)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
63/486,054 21 February 2023 (21.02.2023) US

(71) **Applicant** *(for US only)*: **SCHLUMBERGER TECHNOLOGY CORPORATION** [US/US]; 300 Schlumberger Drive, Sugar Land, Texas 77478 (US).

(71) **Applicant** *(for CA only)*: **SCHLUMBERGER CANADA LIMITED** [CA/CA]; 125 – 9 Avenue SE, Calgary, Alberta T2G 0P6 (CA).

(71) **Applicant** *(for FR only)*: **SERVICES PETROLIERS SCHLUMBERGER** [FR/FR]; 42 rue Saint Dominique, 75007 Paris (FR).

(71) **Applicant** *(for all designated States except CA, FR, US)*: **GEOQUEST SYSTEMS B.V.** [NL/NL]; Parkstraat 83, 2514 JG The Hague (NL).

(72) **Inventor: BENABBOU, Azeddine**; 895 Rue de la Vieille Poste, 34006 Montpellier (FR).

(74) **Agent: MOONEY, Christopher M.** et al.; Schlumberger, 10001 Richmond Avenue, Room 4720, Houston, Texas 77042 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ,

(54) **Title:** OPTIMIZATION FRAMEWORK

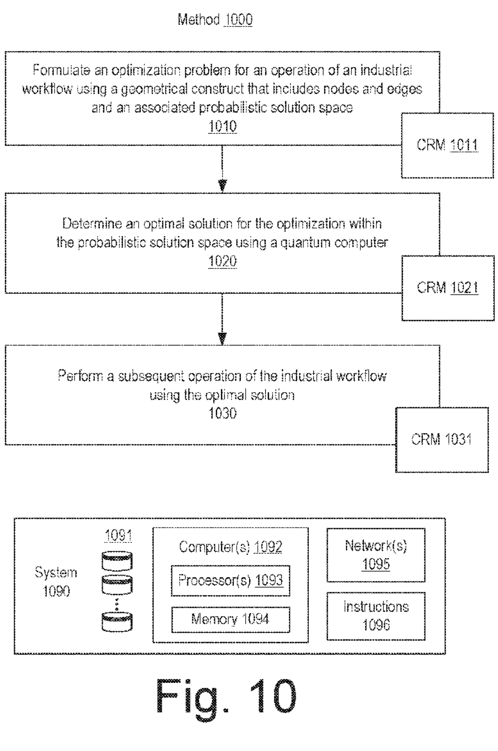


Fig. 10

(57) **Abstract:** A method can include formulating an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space; determining an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and, using the optimal solution, performing a subsequent operation of the industrial workflow.

WO 2024/178046 A1

[Continued on next page]

RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

OPTIMIZATION FRAMEWORK

RELATED APPLICATION

[0001] This application claims priority to and the benefit of a US Provisional Application having Serial No. 63/486,054, filed 21 February 2023, which is incorporated by reference herein in its entirety.

BACKGROUND

[0002] Various types of optimization problems can be challenging to solve using classical computing. For example, various problems that arise in the oil and gas industry can be too complex for practically solving them using classical computing. While approximations may be made to reduce classical computing demands, solutions may not be accurate or verifiable. As an example, consider an operational problem where more than 20 wells are to be drilled in a sequence, each to reach a reservoir. Such a problem can be difficult to solve using classical computing. As another example, consider mesh generation for simulation of physical phenomena. Simulation meshes can involve a large number of points (e.g., nodes) where an optimal mesh is seldom achieved. Simulation meshes generated using classical computing often include ill-shaped elements, ad hoc spacings, etc., which can, at times, confound a simulator's ability to generate simulation results. Various methods, frameworks, systems, etc., described herein can facilitate workflows in the oil and gas and other industries through use of one or more quantum computing techniques.

SUMMARY

[0003] A method can include formulating an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space; determining an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and, using the optimal solution, performing a subsequent operation of the industrial workflow. A system can include a processor; memory accessible by the processor; and processor-executable instructions stored in the memory that are

executable to instruct the system to: formulate an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space; determine an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and, using the optimal solution, perform a subsequent operation of the industrial workflow. One or more computer-readable storage media can include computer-executable instructions executable to instruct a computer to: formulate an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space; determine an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and, using the optimal solution, perform a subsequent operation of the industrial workflow. Various other examples of methods, devices, systems, etc., are also disclosed.

[0004] This summary is provided to introduce a selection of concepts that are further described below in the detailed description. This summary is not intended to identify key or essential features of the claimed subject matter, nor is it intended to be used as an aid in limiting the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] Features and advantages of the described implementations can be more readily understood by reference to the following description taken in conjunction with the accompanying drawings.

[0006] Figure 1 illustrates an example of a framework and an example of a geologic environment;

[0007] Figure 2 illustrates an example of a geometrical construct for an optimization problem;

[0008] Figure 3 illustrates an example of a method;

[0009] Figure 4 illustrates an example of a search space;

[0010] Figure 5 illustrates examples of methods;

[0011] Figure 6 illustrates examples of gate-based techniques;

[0012] Figure 7 illustrates an example of a method;

[0013] Figure 8 illustrates examples of couplers;

[0014] Figure 9 illustrates an example of a quantum computing environment;

[0015] Figure 10 illustrates an example of a method and an example of a system;

[0016] Figure 11 illustrates an example of a computational framework; and

[0017] Figure 12 illustrates example components of a system and a networked system.

DETAILED DESCRIPTION

[0018] The following description includes the best mode presently contemplated for practicing the described implementations. This description is not to be taken in a limiting sense, but rather is made merely for the purpose of describing the general principles of the implementations. The scope of the described implementations should be ascertained with reference to the issued claims.

[0019] Figure 1 shows an example of a system 100 that includes a workspace framework 110 that can provide for instantiation of, rendering of, interactions with, etc., a graphical user interface (GUI) 120. In the example of Figure 1, the GUI 120 can include graphical controls for computational frameworks (e.g., applications) 121, projects 122, visualization 123, one or more other features 124, data access 125, and data storage 126.

[0020] In the example of Figure 1, the workspace framework 110 may be tailored to a particular geologic environment such as an example geologic environment 150. For example, the geologic environment 150 may include layers (e.g., stratification) that include a reservoir 151 and that may be intersected by a fault 153. As an example, the geologic environment 150 may be outfitted with a variety of sensors, detectors, actuators, etc. For example, equipment 152 may include communication circuitry to receive and to transmit information with respect to one or more networks 155. Such information may include information associated with downhole equipment 154, which may be equipment to acquire information, to assist with resource recovery, etc. Other equipment 156 may be located remote from a wellsite and include sensing, detecting, emitting or other circuitry. Such equipment may include storage and communication circuitry to store and to communicate data, instructions, etc. As an example, one or more satellites may be provided for purposes of communications, data acquisition, etc. For example, Figure 1 shows a satellite in communication with the network 155 that may be configured for communications, noting that the satellite may additionally or alternatively include circuitry for imagery (e.g., spatial, spectral, temporal, radiometric, etc.).

[0021] Figure 1 also shows the geologic environment 150 as optionally including equipment 157 and 158 associated with a well that includes a substantially horizontal portion that may intersect with one or more fractures 159. For example, consider a well in a shale formation that may include natural fractures, artificial fractures (e.g., hydraulic fractures) or a combination of natural and artificial fractures. As an example, a well may be drilled for a reservoir that is laterally extensive. In such an example, lateral variations in properties, stresses, etc. may exist where an assessment of such variations may assist with planning, operations, etc. to develop a laterally extensive reservoir (e.g., via fracturing, injecting, extracting, etc.). As an example, the equipment 157 and/or 158 may include components, a system, systems, etc. for fracturing, seismic sensing, analysis of seismic data, assessment of one or more fractures, etc.

[0022] In the example of Figure 1, the GUI 120 shows some examples of computational frameworks, including the DRILLPLAN, PETREL, TECHLOG, PETROMOD, ECLIPSE, INTERSECT, PIPESIM and OMEGA frameworks (SLB, Houston, Texas). As to another type of framework, consider, for example, the DRILLOPS framework. As to yet another type of framework, consider, for example, an emissions framework (EF), which may be operable in combination with one or more other frameworks to make determinations as to emissions (e.g., of one or more field operations, etc.). In such an example, an EF may provide feedback such that another framework can operate on output of the EF, for example, to revise a plan, revise a control scheme, etc., which may be in a manner that aims to reduce one or more types of emissions and/or other impact from an activity, etc.

[0023] The DRILLPLAN framework provides for digital well construction planning and includes features for automation of repetitive tasks and validation workflows, enabling improved quality drilling programs (e.g., digital drilling plans, etc.) to be produced quickly with assured coherency.

[0024] The DRILLOPS framework may execute a digital drilling plan and ensures plan adherence, while delivering goal-based automation. The DRILLOPS framework may generate activity plans automatically for operations, whether they are monitored and/or controlled on the rig or in town. Automation may utilize data analysis and learning systems to assist and optimize tasks, such as, for example, setting rate of penetration (ROP) to drilling a stand. A preset menu of automatable drilling tasks may be rendered, and, using data analysis and models, a plan may be executed in a manner to achieve a specified goal, where, for example, measurements may be utilized for calibration. The DRILLOPS framework provides flexibility to modify and replan activities dynamically, for example, based on a live appraisal of various factors (e.g., equipment, personnel, and supplies). Well construction activities (e.g., tripping, drilling, cementing, etc.) may be continually monitored and dynamically updated using feedback from operational activities. The DRILLOPS framework may provide for various levels of automation based on planning and/or re-planning (e.g., via the DRILLPLAN framework), feedback, etc.

[0025] The PETREL framework can be part of the DELFI cognitive E&P environment (SLB, Houston, Texas) for utilization in geosciences and geoengineering, for example, to analyze subsurface data from exploration to production of fluid from a reservoir.

[0026] The TECHLOG framework can handle and process field and laboratory data for a variety of geologic environments (e.g., deepwater exploration, shale, etc.). The TECHLOG framework can structure wellbore data for analyses, planning, etc.

[0027] The PETROMOD framework provides petroleum systems modeling capabilities that can combine one or more of seismic, well, and geological information to model the evolution of a sedimentary basin. The PETROMOD framework can predict if, and how, a reservoir has been charged with hydrocarbons, including the source and timing of hydrocarbon generation, migration routes, quantities, and hydrocarbon type in the subsurface or at surface conditions.

[0028] The ECLIPSE framework provides a reservoir simulator (e.g., as a computational framework) with numerical solutions for fast and accurate prediction of dynamic behavior for various types of reservoirs and development schemes.

[0029] The INTERSECT framework provides a high-resolution reservoir simulator for simulation of detailed geological features and quantification of uncertainties, for example, by creating accurate production scenarios and, with the integration of precise models of the surface facilities and field operations, the INTERSECT framework can produce reliable results, which may be continuously updated by real-time data exchanges (e.g., from one or more types of data acquisition equipment in the field that can acquire data during one or more types of field operations, etc.). The INTERSECT framework can provide completion configurations for complex wells where such configurations can be built in the field, can provide detailed enhanced-oil-recovery (EOR) formulations where such formulations can be implemented in the field, can analyze application of steam injection and other thermal EOR techniques for implementation in the field, advanced production controls in terms of reservoir coupling and flexible field management, and flexibility to script customized solutions for improved modeling and field management control. The INTERSECT framework, as with the other example frameworks, may be utilized as part of the DELFI cognitive E&P environment (DELFI environment), for example, for rapid simulation of multiple concurrent cases. For example, a workflow may utilize one or more of the DELFI on demand reservoir simulation features.

[0030] The PIPESIM simulator includes solvers that may provide simulation results such as, for example, multiphase flow results (e.g., from a reservoir to a wellhead and beyond, etc.), flowline and surface facility performance, etc. The PIPESIM simulator may be integrated, for example, with the AVOCET production operations framework (SLB, Houston Texas). As an example, a reservoir or reservoirs may be simulated with respect to one or more enhanced recovery techniques (e.g., consider a thermal process such as steam-assisted gravity drainage (SAGD), etc.). As an example, the PIPESIM simulator may be an optimizer that can optimize one or more operational scenarios at least in part via simulation of physical phenomena.

[0031] The OMEGA framework includes finite difference modelling (FDMOD) features for two-way wavefield extrapolation modelling, generating synthetic shot gathers with and without multiples. The FDMOD features can generate synthetic shot gathers by using full 3D, two-way wavefield extrapolation modelling, which can utilize wavefield extrapolation logic matches that are used by reverse-time migration (RTM). A model may be specified on a dense 3D grid as velocity and optionally as anisotropy, dip, and variable density. The OMEGA framework also includes features for RTM, FDMOD, adaptive beam migration (ABM), Gaussian packet migration (Gaussian PM), depth processing (e.g., Kirchhoff prestack depth migration (KPSDM), tomography (Tomo)), time processing (e.g., Kirchhoff prestack time migration (KPSTM), general surface multiple prediction (GSMP), extended interbed multiple prediction (XIMP)), framework foundation features, desktop features (e.g., GUIs, etc.), and development tools. Various features can be included for processing various types of data such as, for example, one or more of: land, marine, and transition zone data; time and depth data; 2D, 3D, and 4D surveys; isotropic and anisotropic (TTI and VTI) velocity fields; and multicomponent data.

[0032] The aforementioned DELFI environment provides various features for workflows as to subsurface analysis, planning, construction and production, for example, as illustrated in the workspace framework 110. As shown in Figure 1, outputs from the workspace framework 110 can be utilized for directing, controlling, etc., one or more processes in the geologic environment 150 and, feedback 160, can be received via one or more interfaces in one or more forms (e.g., acquired data as to operational conditions, equipment conditions, environment conditions, etc.).

[0033] As an example, a workflow may progress to a geology and geophysics ("G&G") service provider, which may generate a well trajectory, which may involve execution of one or more frameworks. A framework may operatively couple various other frameworks to provide for a multi-framework workspace. As an example, the GUI 120 of Figure 1 may be a GUI of the DELFI environment, which can be considered a multi-framework environment or a framework of frameworks.

[0034] In the example of Figure 1, the visualization features 123 may be implemented via the workspace framework 110, for example, to perform tasks as associated with one or more of subsurface regions, planning operations, constructing wells and/or surface fluid networks, and producing from a reservoir.

[0035] As an example, visualization features can provide for visualization of various earth models, properties, etc., in one or more dimensions. As an example, visualization features can provide for rendering of information in multiple dimensions, which may optionally include multiple resolution rendering. In such an example, information being rendered may be associated with one or more frameworks and/or one or more data stores. As an example, visualization features may include one or more control features for control of equipment, which can include, for example, field equipment that can perform one or more field operations. As an example, a workflow may utilize one or more frameworks to generate information that can be utilized to control one or more types of field equipment (e.g., drilling equipment, wireline equipment, fracturing equipment, etc.).

[0036] As to a reservoir model that may be suitable for utilization by a simulator, consider acquisition of seismic data as acquired via reflection seismology, which finds use in geophysics, for example, to estimate properties of subsurface formations. As an example, reflection seismology may provide seismic data representing waves of elastic energy (e.g., as transmitted by P-waves and S-waves, in a frequency range of approximately 1 Hz to approximately 100 Hz). Seismic data may be processed and interpreted, for example, to understand better composition, fluid content, extent and geometry of subsurface rocks. Such interpretation results can be utilized to plan, simulate, perform, etc., one or more operations for production of fluid from a reservoir (e.g., reservoir rock, etc.).

[0037] Field acquisition equipment may be utilized to acquire seismic data, which may be in the form of traces where a trace can include values organized with respect to time and/or depth (e.g., consider 1D, 2D, 3D or 4D seismic data). For example, consider acquisition equipment that acquires digital samples at a rate of one sample per approximately 4 ms. Given a speed of sound in a medium or media, a sample rate may be converted to an approximate distance. For example, the speed of sound in rock may be on the order of around 5 km per second. Thus, a sample time spacing of approximately 4 ms would correspond to a sample "depth" spacing of about 10 meters (e.g., assuming a path length from source to boundary and boundary to sensor). As an example, a trace may be about 4 seconds in duration; thus, for a sampling rate of one sample at about 4 ms intervals, such a trace would include about 1000 samples where latter acquired samples correspond to deeper reflection boundaries. If the 4 second trace duration of the foregoing example is divided by two (e.g., to account for reflection), for a vertically aligned source and sensor, a deepest boundary depth may be estimated to be about 10 km (e.g., assuming a speed of sound of about 5 km per second).

[0038] As an example, a model may be a simulated version of a geologic environment. As an example, a simulator may include features for simulating physical phenomena in a geologic environment based at least in part on a model or models. A simulator, such as a reservoir simulator, can simulate fluid flow in a geologic environment based at least in part on a model that can be generated via a framework that receives seismic data. A simulator can be a computerized system (e.g., a computing system) that can execute instructions using one or more processors to solve a system of equations that describe physical phenomena subject to various constraints. In such an example, the system of equations may be spatially defined (e.g., numerically discretized) according to a spatial model that that includes layers of rock, geobodies, etc., that have corresponding positions that can be based on interpretation of seismic and/or other data. A spatial model may be a cell-based model where cells are defined by a grid (e.g., a mesh). A cell in a cell-based model can represent a physical area or volume in a geologic environment where the cell can be assigned physical properties (e.g., permeability, fluid properties, etc.) that may be germane to one or more physical phenomena (e.g., fluid volume, fluid flow, pressure, etc.). A reservoir simulation model can be a spatial model that may be cell-based.

[0039] A simulator can be utilized to simulate the exploitation of a real reservoir, for example, to examine different productions scenarios to find an optimal one before production or further production occurs. A reservoir simulator does not provide an exact replica of flow in and production from a reservoir at least in part because the description of the reservoir and the boundary conditions for the equations for flow in a porous rock are generally known with an amount of uncertainty. Certain types of physical phenomena occur at a spatial scale that can be relatively small compared to size of a field. A balance can be struck between model scale and computational resources that results in model cell sizes being of the order of meters; rather than a lesser size (e.g., a level of detail of pores). A modeling and simulation workflow for multiphase flow in porous media (e.g., reservoir rock, etc.) can include generalizing real micro-scale data from macro scale observations (e.g., seismic data and well data) and upscaling to a manageable scale and problem size. Uncertainties can exist in input data and solution procedure such that simulation results too are to some extent uncertain. A process known as history matching can involve comparing simulation results to actual field data acquired during production of fluid from a field. Information gleaned from history matching, can provide for adjustments to a model, data, etc., which can help to increase accuracy of simulation.

[0040] As an example, a simulator may utilize various types of constructs, which may be referred to as entities. Entities may include earth entities or geological objects such as wells, surfaces, reservoirs, etc. Entities can include virtual representations of actual physical entities that may be reconstructed for purposes of simulation. Entities may include entities based on data acquired via sensing, observation, etc. (e.g., consider entities based at least in part on seismic data and/or other information). As an example, an entity may be characterized by one or more properties (e.g., a geometrical pillar grid entity of an earth model may be characterized by a porosity property, etc.). Such properties may represent one or more measurements (e.g., acquired data), calculations, etc.

[0041] While several simulators are illustrated in the example of Figure 1, one or more other simulators may be utilized, additionally or alternatively. For example, consider the KINETIX/VISAGE geomechanics simulator (SLB, Houston Texas). The KINETIX/VISAGE simulator can include finite element numerical solvers that may provide simulation results such as, for example, results as to compaction and subsidence of a geologic environment, well and completion integrity in a geologic environment, cap-rock and fault-seal integrity in a geologic environment, fracture behavior in a geologic environment, thermal recovery in a geologic environment, CO_2 disposal, etc. The aforementioned PIPESIM simulator includes solvers that may provide simulation results such as, for example, multiphase flow results (e.g., from a reservoir to a wellhead and beyond, etc.), flowline and surface facility performance, etc. The PIPESIM simulator may be integrated, for example, with the AVOCET production operations framework (SLB, Houston Texas). As an example, a reservoir or reservoirs may be simulated with respect to one or more enhanced recovery techniques (e.g., consider a thermal process such as steam-assisted gravity drainage (SAGD), etc.). As an example, the PIPESIM simulator may be an optimizer that can optimize one or more operational scenarios at least in part via simulation of physical phenomena. The MANGROVE simulator (SLB, Houston, Texas) provides for optimization of stimulation design (e.g., stimulation treatment operations such as hydraulic fracturing) in a reservoir-centric environment. The MANGROVE framework can combine scientific and experimental work to predict geomechanical propagation of hydraulic fractures, reactivation of natural fractures, etc., along with production forecasts within 3D reservoir models (e.g., production from a drainage area of a reservoir where fluid moves via one or more types of fractures to a well and/or from a well). The MANGROVE framework can provide results pertaining to heterogeneous interactions between hydraulic and natural fracture networks, which may assist with optimization of the number and location of fracture treatment stages (e.g., stimulation treatment(s)), for example, to increased perforation efficiency and recovery.

[0042] Various workflows can utilize one or more frameworks where such workflows can involve one or more optimizations. For example, the DRILLPLAN framework may involve optimizing a sequence for drilling of wells to reach a reservoir. As explained, various frameworks can utilize meshes, which may be geometrical constructs for discretization of equations that describe physical phenomena. Optimization of such meshes can facilitate workflows, for example, consider a mesh of suitably shaped elements that can help to reduce simulation errors and increase convergence of an iterative solver.

[0043] Combinatorial optimization can involve problems of optimizing an objective function subject to a number of constraints. In various applications, the objective function of the underlying optimization problem involves a small number of variables, and the constraints are induced by a family of geometric objects. Such problems can be referred to as geometrical optimization problems. In such cases one may expect that the underlying geometry can be exploited to obtain faster and simpler algorithms.

[0044] Combinatorial optimization can be defined as a process that involves searching for maxima (or minima) of an objective function F whose domain is a discrete but large configuration space (as opposed to an N-dimensional continuous space). Often, the space of possible solutions is too large to search exhaustively using pure brute force. In some cases, problems can be solved exactly using Branch and Bound techniques. However, in other cases no exact algorithms are feasible, and randomized search algorithms are employed.

[0045] As an example of a problem, consider an article by Wang et al., “Efficient Optimization of Well-Drilling Sequence with Learned Heuristics” (SPE J. 24 (2019): 2111–2134. doi: https://doi.org/10.2118/195640-PA), which is incorporated by reference herein. In the Wang article, the problem involves development of a new resource field or the expansion of an existing resource field where estimates are generated prior to making a decision to proceed with drilling of wells. Inputs can include number of wells, well locations, and well types, as well as parameters describing a reservoir to be accessed by the wells. The order in which the wells are drilled can be a relevant contributor to viable development or expansion; hence, optimization of the sequence of drilling wells can be desirable. A naïve way to optimize the drilling sequence would be to simulate each possible sequence and then choose the sequence that gives the best outcome. However, such an approach is classically infeasible when the number of wells that must be sequenced exceeds six. Classical efforts to improve solving such a problem have employed techniques such as simulated annealing to optimize both well locations and drilling order. However, again, the number of wells is practically limited to a relatively small number. For example, in a 12 well problem that employed classical simulated annealing, 100,000 reservoir–simulation runs were required. Such a problem has scale issues thereby making a solution increasingly impractical as the number of wells increases.

[0046] As mentioned, a combinatorial optimization problem can be a geometrical optimization problem. A geometrical optimization problem can involve use of geometric constructs and associated techniques. For example, consider Delaunay triangulation and Voronoi diagrams.

[0047] In mathematics and computational geometry, a Delaunay triangulation (DT) for a given set P of discrete points in a general position is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). Delaunay triangulations maximize the minimum of all the angles of the triangles in the triangulation; they tend to avoid sliver triangles. Sliver triangles may be characterized as triangles with aspect ratios that may be relatively large (e.g., above a desired value), as triangles with a relatively small internal angle (e.g., less than a desired value), as triangles with a relatively large total edge length (e.g., above a desired value), etc.

[0048] For a set of points on the same line there is no Delaunay triangulation (the notion of triangulation is degenerate for this case). For four or more points on the same circle (e.g., the vertices of a rectangle) the Delaunay triangulation is not unique: each of the two possible triangulations that split the quadrangle into two triangles satisfies the “Delaunay condition”, i.e., the requirement that the circumcircles of all triangles have empty interiors.

[0049] By considering circumscribed spheres, the notion of Delaunay triangulation extends to three and higher dimensions. Generalizations are possible to metrics other than Euclidean distance. However, in these cases a Delaunay triangulation is not guaranteed to exist or be unique.

[0050] As to Dirichlet-Voronoi diagrams (or simply Voronoi diagrams), the Delaunay triangulation of a discrete point set P in general position corresponds to the dual graph of the Voronoi diagram for P where the circumcenters of the Delaunay triangles are the vertices of the Voronoi diagram. In a 2D case, the Voronoi vertices are connected via edges, that can be derived from adjacency-relationships of the Delaunay triangles: If two triangles share an edge in the Delaunay triangulation, their circumcenters are to be connected with an edge in the Voronoi tessellation.

[0051] However, there are some special cases where this relationship does not hold, or is ambiguous, which can include cases like: (i) three or more collinear points, where the circumcircles are of infinite radii; (ii) four or more points on a perfect circle, where the triangulation is ambiguous and all circumcenters are trivially identical; and (iii) edges of the Voronoi diagram going to infinity are not defined by this relation in case of a finite set P such that, if the Delaunay triangulation is computed using the Bowyer–Watson algorithm, then the circumcenters of triangles having a common vertex with the "super" triangle are to be ignored and edges going to infinity are to start from a circumcenter and they are perpendicular to the common edge between the kept and ignored triangle.

[0052] As to Dirichlet triangulation (DT), the DT of a point set is defined as the dual of the Voronoi diagrams of the set. The 2D-DT is formed by connecting two points if and only if their Voronoi regions have a common border segment. If no four or more points are cocircular, the vertices of the Voronoi diagrams are circumcenters of the Delaunay triangles. This is true because vertices of the Voronoi represent locations that are equidistant to three (or more) sites. Because edges of the Voronoi diagrams are the loci of points equidistant to two sites, each edge of the Voronoi diagrams is perpendicular to the corresponding edge of the DT. This duality extends straightforwardly to 3D. Voronoi diagrams can be considered as using an elegant characteristic from the DT: Voronoi volumes have edges orthogonal to the line segment between adjacent vertices and the intersection point is on the mean point of the edge that connects the two vertices.

[0053] In geometric constructs, concepts of energy can be introduced. For example, consider Dirichlet energy, which can be used to measure variability of a smooth function. Dirichlet energy is associated with the Laplace-Beltrami operator, where either or both can be utilized in geometrical processing. The Laplace-Beltrami operator is utilized in differential geometry as a generalization of the Laplace operator to functions defined on submanifolds in Euclidean space and, even more generally, on Riemannian and pseudo-Riemannian manifolds. As an example, a geometrical optimization can involve using concepts of energy, for example, to maximize energy or to minimize energy, which may provide a solution to a problem.

[0054] As explained, classical approaches to geometrical optimization problems can be, practically, quite limited. Such problems tend to scale enormously with respect to an increasing number of points.

[0055] As an example, a geometrical optimization problem may be cast in terms of quantum mechanics such that a solution may be achieved or more readily achieved when compared to a classical approach.

[0056] Quantum mechanics provides a description of physical properties of nature at the scale of atoms and subatomic particles and serves as a foundation for quantum physics including quantum chemistry, quantum field theory, quantum technology, and quantum information science. Classical physics describes many aspects of nature at an ordinary (macroscopic) scale, while proving insufficient for a description at small (atomic and subatomic) scales. Quantum mechanics differs from classical physics in that energy, momentum, angular momentum, and other quantities of a bound system are restricted to discrete values (quantization); objects have characteristics of both particles and waves (wave–particle duality); and there are limits to how accurately the value of a physical quantity can be predicted prior to its measurement, given a complete set of initial conditions (the uncertainty principle).

[0057] For classical mechanics and quantum mechanics, two concepts may be utilized. A first concept considers the complete state of a mechanical system at a given time, as may be encoded as a phase point (classical mechanics) or a pure quantum state vector (quantum mechanics) while a second concept considers an equation of motion which carries the state forward in time: Hamilton's equations (classical mechanics) or the Schrödinger equation (quantum mechanics). Using these two concepts, the state at any other time, past or future, can in principle be computed. There is however a disconnect between these laws and everyday life experiences, as it may not be necessary (nor even theoretically possible) to know exactly at a microscopic level the simultaneous positions and velocities of each molecule while carrying out processes at a particular scale (for example, when performing a chemical reaction). Statistical mechanics can fill this disconnection between the laws of mechanics and the practical experience of incomplete knowledge, by adding some uncertainty about which state the system is in.

[0058] Whereas ordinary mechanics only considers the behavior of a single state, statistical mechanics introduces the statistical ensemble, which is a large collection of independent copies (e.g., virtual copies) of the system in various states. The statistical ensemble, for some examples, is a probability distribution over all possible states of the system. In classical statistical mechanics, the ensemble is a probability distribution over phase points (as opposed to a single-phase point in ordinary mechanics), usually represented as a distribution in a phase space with canonical coordinate axes. In quantum statistical mechanics, the ensemble, for some examples, is a probability distribution over pure states, and can be compactly summarized as a density matrix.

[0059] As is usual for probabilities, an ensemble can be interpreted in different ways. For example, an ensemble can be taken to represent the various possible states that a single system could be in (epistemic probability, a form of knowledge), or the members of the ensemble can be understood as the states of the systems in experiments repeated on independent systems which have been prepared in a similar but imperfectly controlled manner (empirical probability), in the limit of an infinite number of trials.

[0060] Regardless of how probability is interpreted, each state in an ensemble evolves over time according to the equation of motion. Thus, the ensemble itself (the probability distribution over states) also evolves, for example, as the virtual systems in the ensemble continually leave one state and enter another. The ensemble evolution is given by the Liouville equation (classical mechanics) or the von Neumann equation (quantum mechanics). These equations, for some examples, are derived by the application of the mechanical equation of motion separately to each system contained in the ensemble, with the probability of the system being conserved over time as it evolves from state to state.

[0061] A Quantum Approximate Optimization Algorithm includes a quantum algorithm (QAOA) that produces approximate solutions for combinatorial optimization problems. The algorithm depends on an integer $p \geq 1$ and the quality of the approximation improves as p is increased. A quantum circuit that implements the algorithm can include unitary gates whose locality is at most the locality of the objective function whose optimum is sought. The depth of the circuit grows linearly with p-times (at worst) the number of constraints. If p is fixed, that is, independent of the input size, the algorithm makes use of efficient classical preprocessing.

[0062] As an example, a method includes using quantum approximate optimization. As explained, for various combinatorial optimization problems, finding the exact optimal solution can be NP-complete. There are also hardness-of-approximation results proving that finding an approximation with sufficiently small error bound is NP-complete. For certain problems there is a gap between the best error bound achieved by a polynomial-time classical approximation algorithm and the error bound proven to be NP-hard. The aforementioned QAOA was proven to solve a combinatorial optimization problem called Max E3LIN2 with a better approximation ratio than any polynomial-time classical algorithm known at the time. The Max E3LIN2 problem is a combinatorial problem of bounded occurrences; given a set of linear equations containing exactly three Boolean variables (E3) which sum to 0 or 1 mod 2 (LIN2), find a solution which maximizes the number of satisfied solutions. Each variable is guaranteed to be in no more than D equations.

[0063] As explained, for various industrial problems, it can be quite difficult to obtain optimal solutions using classical techniques. As such, a method can employ quantum computing to solve one or more geometrical optimization problems. For example, consider an approach that involves generation of a Delaunay triangulation and more generally a Dirichlet triangulation in n dimensions. Such an approach can be generalized where, for example, a method can include generating one or more constrained Delaunay triangulations, etc., (e.g., meshes) that can be used in one or more numerical simulations. As an example, a quantum approach can be applied to a well optimization problem where, for example, a number of wells are to be drilled where the number of wells can exceed 10 wells and may exceed 20 wells (e.g., consider more than 30 wells).

[0064] As an example, a quantum approach to geometrically cast problems can be employed to solve various problems that exist or may arise in the energy industry. As mentioned, a well scheduling problem is one type of problem that may be expressed as a Delaunay optimization type of problem. For example, well sites can correspond to point positions where scheduling is cast as an insertion sequence of points in a Delaunay triangulation. In such an example, finding the optimal insertion sequence in the triangulation is equivalent to finding the optimal well scheduling (which can be crucial for the net present value NPV in when preparing a field-development plan).

[0065] As an example, a computational framework can include various features to implement one or more quantum computing techniques. Such a framework may be suitable for handling problems in a manner that can guarantee that an optimal solution is obtained. For example, a solution given via quantum computing can corresponds to a ground state of a physical system of particles. In such an example, via the principal of minimum energy, the system of particles can be assured of convergence to its ground state and hence represent the optimal solution of a problem. As an example, a framework can implement one or more quantum computing techniques to solve one or more types of problems for which, at present, no classical algorithm exists. For example, consider a Dirichlet triangulation quantum computing approach that minimizes the Dirichlet energy of an arbitrary piece-wise linear function. As an example, a framework can be extensible to handle various scales. For example, the optimal solution for a Delaunay triangulation problem may be achieved by exploring an entire set of N! (N factorial) insertion sequences, where N is the number of points in the triangulation. Such a problem is impractical using classical computing even for moderate N (e.g., for N=100 N! is $\sim 10^{157}$, which is larger than number of particles in the universe).

[0066] As an example, a framework can implement one or more quantum computing techniques for mesh generation. Mesh generation is a task that finds use in simulations, visualizations, etc. For example, a mesh can be generated for discretization of equations such as partial differential equations that represent physical phenomena (e.g., transport phenomena, etc.). As to visualizations, meshes can be utilized to create virtual and/or augmented scenes. For example, consider a mesh that represents an actor in a game, a movie, a training video, etc., where such a mesh can be generated using points (e.g., triangulation using points, etc.).

[0067] Mesh generation can involve finding optimal point positions such that any triangle (e.g., or tetrahedron) formed by three (e.g., or four) points has the best shape possible, which, as explained in some examples, is characterized using one or more metrics (e.g., aspect ratio, internal angle, sum of lengths, etc.). This form of criterion (e.g., well-shaped mesh elements) impacts convergence and result quality of a mesh-based numerical simulation. Convergence may be achieved in an expedited manner using fewer iterations and hence lesser computational resources (e.g., compute and/or time) while fewer iterations may also help to reduce various types of error such as digital roundoff error. Other mesh generation tasks in some examples involve determining triangles for fixed points. As an example, a mesh generation task may involve determining triangles for fixed points and movable points. Mesh generation can be cast in various terms where points and edges can be utilized for purposes of forming geometric shapes such as, for example, triangles. As explained, mesh generation can be a geometrical optimization problem, which, as mentioned, can be solved at least in part using one or more quantum computing techniques.

[0068] As an example, a framework can provide for formulating and solving one or more geometrical optimization problems using, at least in part, one or more quantum computing techniques. For example, consider formulating a geometrical optimization problem as a minimization of a cost function under constraints. As to an example of a quantum computing technique, consider quantum annealing, which may be utilized for problem formulation and solution.

[0069] Quantum annealing can be considered a type of adiabatic quantum computation that involves an initial Hamiltonian whose ground state is relatively straight forward to prepare and that slowly varies the Hamiltonian to one whose ground state encodes the solution to a computational problem. By the adiabatic theorem, the system may track the instantaneous ground state provided the variation of the Hamiltonian is sufficiently slow. The runtime of an adiabatic algorithm scales at worst as $1/\gamma^3$ where γ is the minimum eigenvalue gap between the ground state and the first excited state. If the Hamiltonian is varied sufficiently smoothly, the scaling can improve to $\tilde{O}(1/\gamma^2)$. Adiabatic quantum computation may be utilized for solving NP-complete combinatorial optimization problems. Adiabatic quantum algorithms for optimization problems may utilize "stoquastic" Hamiltonians, which do not suffer from the sign problem. Such algorithms are sometimes referred to as quantum annealing. Adiabatic quantum computation with non-stoquastic Hamiltonians can be as powerful as the quantum circuit model. Adiabatic algorithms using stoquastic Hamiltonians may be less powerful yet likely more powerful than classical computation. As an example, quantum annealing may be implemented with reference to a classical optimization algorithm that works by simulating a quantum process, much as simulated annealing is a classical optimization algorithm that works by simulating a thermal process.

[0070] As explained, quantum annealing can be implemented as part of an optimization process for finding a solution such as, for example, a global minimum, of a given objective function over a given set of candidate solutions (candidate states), by a process using quantum fluctuations. Quantum annealing can be used for problems where a search space is discrete (e.g., combinatorial optimization problems) with various local minima. Quantum annealing can start from a quantum-mechanical superposition of all possible states (candidate states) with equal weights. Then the system can evolve following the time-dependent Schrödinger equation, a natural quantum-mechanical evolution of physical systems. The amplitudes of all candidate states keep changing, realizing a quantum parallelism, according to the time-dependent strength of the transverse field, which causes quantum tunneling between states. If the rate of change of the transverse field is slow enough, the system stays close to the ground state of the instantaneous Hamiltonian. If the rate of change of the transverse field is accelerated, the system may leave the ground state temporarily but produce a higher likelihood of concluding in the ground state of the final problem Hamiltonian, i.e., diabatic quantum computation. The transverse field can be finally switched off, and the system is expected to have reached the ground state of the classical Ising model that corresponds to the solution to the original optimization problem.

[0071] As an example, a framework can utilize a quantum formulation for various geometrical problems. For example, consider formulating and solving of Delaunay triangulation and constrained Delaunay triangulation in n dimensions using quantum annealing and/or formulating and solving of mesh generation with the highest quality possible in n dimensions using quantum annealing.

[0072] As explained, a framework can implement one or more quantum computing techniques for one or more geometrical optimization problems, which can involve generation of constrained triangulations and meshes. Below, an example is presented that involves formulation of a constrained Delaunay triangulation problem where the formulation can be generalized using one or more other triangulations, meshes, etc.

[0073] For application of quantum computing techniques, a triangulation can be described as a minimization problem. For a Delaunay triangulation, energy to be minimized can be formulated to correspond to a volume under a convex hull described by an equation, which in two-dimensions can be given as $z = x^2 + y^2$.

[0074] Figure 2 shows an example of a diagram 200 of an example of a geometrical structure in a multidimensional space where nodes (e.g., points) and edges can be projected to reduce the dimension of the geometrical structure. In the example of Figure 2, for sake of visualization, the geometrical structure is presented in a three-dimensional space and as a projection to a two-dimensional space. In the example of Figure 2, one of the triangles as defined by three nodes and three edges are highlighted along with dashed lines indicating a correspondence between the spaces.

[0075] The energy to minimize in the Delaunay case can be written as: $E = \sum_{t_i} vol_i$, where t_i and vol_i are triangle of index *i* and its corresponding volume (e.g., the volume under the convex hull of the parabolic lifting map). To compute a valid triangulation, a method can include add the following two constraints:

- constraint C1: the number of triangles (e.g., or tetrahedra) adjacent to border edges (triangles) is to be 1.
- constraint C2: the number of triangles (e.g., or tetrahedra) adjacent to internal edges (triangles) is to be 2.

[0076] As explained, use of a 3D structure and a 2D projection can facilitate visualization, noting that the formulation can be applied to higher dimensions.

[0077] As to input, consider n points (e.g., nodes) and m edges. Given the input, consider formulating a problem as a binary quadratic model (BQM). In such a formulation, variables are the edges, where each edge is a combination of two different points. In such an example, each edge can be represented by a qubit and can have two values: 1 if the edge is present in the Delaunay triangulation and 0 if it is not present in the Delaunay triangulation. In this formulation, a triangle is implicitly represented as a combination of two edges that share one and only one point.

[0078] Above, the term qubit is utilized, which is a quantum bit. A qubit can be considered to be a quantum computing counterpart to a binary digit or bit of classical computing. Just as a bit is the basic unit of information in classical computing, a qubit, for some examples, is the basic unit of information in a quantum computing.

[0079] As to output, a solution for the foregoing formulated problem, for some examples, is a vector of qubits (edges) that minimizes energy E and satisfies the constraints C1 and C2.

[0080] Figure 3 shows an example of a method 300 with input of points 0, 1, 2 and 3 and border edges 01, 12, 23, and 30 along with output, which is a triangulation (e.g., a solution to a minimization problem). The input can be a relatively simple input for Delaunay triangulation in 2D. For the example of Figure 3, there are 6 possible edges (qubit variables): q_0=01, q_1=12, q_2=23, q_3=02, q_4=13, and q_5=03. In general, the number of variables is: $\frac{n^2-n}{2}$ (where n is the number of points).

[0081] Table 1. Weight w_{ij} that is associated to each edge combination $q_i q_j$.

qubit(edge)	q_0=01	q_1=12	q_2=23	q_3=02	q_4=13	q_5=03
q_0=01	0	1	0	1	1	1
q_1=12	1	0	1	1	1	0
q_2=23	0	1	0	1	1	1
q_3=02	1	1	1	0	0	1
q_4=13	1	1	1	0	0	1
q_5=03	1	0	1	1	1	0

[0082] Table 1, above, shows possible triangles for the example of Figure 3 along with means possible and means not possible (e.g., triangle 01-12 is possible because edge 01 and 12 are sharing point 1, but triangle 01-23 is not a possible since the two edges do not share any point).

[0083] The matrix in Table 1 is symmetric, e.g., triangle 12-01= triangle 01-12, and the zero value along the diagonal expresses the fact that the combination of an edge with itself is not a triangle. Also note that some edge combinations are equivalent (i.e., they give the same triangle, e.g., triangle 01-12 = triangle 01-02 and triangle 01-13 = triangle 01-03).

[0084] As an example, a minimization problem can be written in terms of qubit variables as:

$$\min\left(\sum_{i}\sum_{j\neq i} w_{ij} \times v_{ij} \times q_i q_j\right)$$

where w_{ij} is the entry in Table 1 for triangle $q_i q_j$ and v_{ij} is its volume.

[0085] Below, various constraints are set forth:

$q_0 + q_1 + q_2 + q_5 = 4$, existing input edges

$\frac{\sum_{j\neq 0} w_{0j} \times q_0 q_j}{2} = 1$, constraint C1 for q_0

$\frac{\sum_{j\neq 1} w_{1j} \times q_1 q_j}{2} = 1$, constraint C1 for q_1

$\frac{\sum_{j\neq 2} w_{2j} \times q_2 q_j}{2} = 1$, constraint C1 for q_2

$\frac{\sum_{j\neq 5} w_{5j} \times q_5 q_j}{2} = 1$, constraint C1 for q_5

$\left(\frac{\sum_{j\neq 3} w_{3j} \times q_3 q_j}{2}\right)\%2 = 0$, constraint C2 for q_3

$\left(\frac{\sum_{j\neq 4} w_{4j} \times q_4 q_j}{2}\right)\%2 = 0$, constraint C2 for q_4

[0086] The solution with minimal energy, given by a quantum machine, is the vector: (1 1 1 1 0 1). Figure 3 shows the solution graphically (e.g., geometrically), which corresponds to the Delaunay triangulation of the given points. In other words, the resulting output or solution is a Delaunay triangulation.

[0087] In the example of Figure 3, as mentioned, a quadratic model can be utilized such as for example:

$$\text{minimize: } \sum_{i}\sum_{j>i} m_{ij} vol_{ij} e_i e_j$$

subject to:

$$e_0 + e_1 + e_2 + e_5 = 4$$

$$\textstyle\sum_{j\neq 3} m_{3j} a_{3j} e_{3e_j} + \sum_{j\neq 4} m_{4j} a_{4j} e_{4e_j} = 2a$$

[0088] In such an approach, a Hamiltonian matrix H_f (control) can be written as:

$$H_f = \begin{bmatrix} h_{00} & \cdots & h_{05} \\ \vdots & \ddots & \vdots \\ & \cdots & h_{55} \end{bmatrix}$$

where h_{ij} is given by the foregoing equations.

[0089] In such an approach, solutions can be eigenvectors (eigenstates):

$$H_f u_i = \alpha_i u_i$$

$$u_i = \textstyle\sum_j w_j e_i, \;\; w_j = 0 \; or \;\; 1$$

[0090] In such an approach, a method can find u_i with the lowest eigenvalue α_i (energy).

[0091] Figure 4 shows an example of a search space 400 for the formulated problem of Figure 3 where the search space can be represented geometrically using superposition. In the example of Figure 4, note that dashed lines correspond to zero while solid lines correspond to 1.

[0092] Figure 5 shows an example of a method 500 for minimization that can involve filtering using the constraints 1 and 2 to arrive at the optimal solution with minimal energy and also shows an example of a method 550 that can include generating an optimal solution 552, setting one or more constraints 554 (e.g., one or more additional constraints), and generating a revised optimal solution 556 subject to the one or more constraints.

[0093] A platform such as, for example, the D-Wave Systems Inc. platform (Burnaby, BC, Canada) may be utilized to perform an optimization such as, for example, a quadratic optimization. The D-Wave platform includes various application programming interfaces (APIs) such as, for example, cloud-based APIs, an API referred to as dimod, which is a shared API for samplers that provides classes for quadratic models (e.g., such as the binary quadratic model (BQM) class that contains Ising and quadratic unconstrained binary optimization (QUBO) models used by samplers, and higher-order (non-quadratic) models), etc.

[0094] Many real-world problems include constraints that can be represented in one or more constrained models. For example, the D-Wave platform includes a constrained quadratic model that can support constraints by encoding both an objective and its set of constraints, as models or in symbolic form.

[0095] Unconstrained quadratic models may be used to submit problems to samplers such as D-Wave quantum computers and some hybrid quantum-classical samplers. When using such samplers to handle problems with constraints, a formulation can involve formulating constraints as penalties, noting that constrained models, such as the constrained quadratic model, can support constraints natively.

[0096] D-Wave quantum computers can accept unconstrained binary quadratic models, such as QUBO models: binary because variables are represented by qubits that return two states and quadratic because polynomial terms of two variables can be represented by pairs of coupled qubits.

[0097] D-Wave Systems Inc. provides quantum computing resources, which may be gate-based or quantum annealing-based. Various quantum computing resources can include a number of qubits in excess of 100, in excess of 1000, etc. For example, D-Wave Systems Inc. has quantum computing resources with 1000s of qubits (e.g., in excess of 5000 qubits). D-Wave Systems Inc.'s Advantage quantum computer includes a processor architecture with over 5000 qubits and 15-way qubit connectivity.

[0098] As an example, a 5000 qubits quantum computer may be suitable for handling a problem with 1 million variables with 100 thousand constraints. For a problem formulated using triangles, the number of variables can scale to the power of three. For example, for 10 nodes (e.g., points), the number of variables can be of the order of 1000. For 100 nodes (e.g., points), the number of variables can be of the order of 1 million. In such an example, the nodes can be geometric nodes where a problem may be an optimization problem for a mesh (e.g., a simulation mesh and/or a visualization mesh), a sequence of events (e.g., drilling of wells), placement of equipment (e.g., placement of wells, etc.). In various instances, for purposes of error, noise, etc., additional qubits can be utilized such that a problem formulated for 1000 qubits may utilize more than 1000 qubits.

[0099] Some hybrid quantum-classical samplers accept constrained and non-binary models; for example, a quadratic model with an integer variable that must be smaller than some configured value.

[0100] Below, an example of code is presented as may be utilized with the D-Wave platform.

[0101] Example Code

```
#instantiate a cqm
cqm=ConstrainedQuadraticModel()

#define the objective function
objective1=quicksum(m[i][j]*vol[i][j]*bin_variables[i]*bin_variables[j] for i in
range(nb_var) for j in range(1+1, nb_var))

#set objectives
cqm.set_objective(objective1)
#add constraints
#constraint 1- existing edges
cqm.add_constraint(bin_variables[0]+bin_variables[1)+
bin_variables[2]+bin_variables[5]==4,
```

```
'existing edge constraint')
#constraint2 = sum of incident areas of incident triangles to internal
# edges - total area (for all edges in the solution)
cqm.add_constraint(bin_variables[3]*quicksum(a[3][1] for i in range(nb_var))+
bin_variables[4]*quicksum(a[4][i] for i in range(nb_var))==2.*total_area, 'total
area')

# submit to a solver, CQM sampler
# instantiate a cqm solver (sampler)
cqm_sampler - LeapHybridCQMSampler()

# # #solve with the instance of the cqm solver
sampleset = cqm_sampler.sample_cqm(cqm, time_limit=5,
label ='Delaunay 2d with objective and constraints')
```

[0102] Table 2. Example Output.

e0	e1	e2	e3	e4	e5	Energy	Occur	Feasible
1	1	1	0	0	1	10.7	6	False
1	0	1	0	1	1	11.9	5	False
1	1	0	0	1	1	12.2	2	False
1	0	1	1	0	1	12.4	1	False
1	1	0	1	0	1	12.9	1	False
1	**1**	**1**	**1**	**0**	**1**	**21.4**	**35**	**True**
1	1	1	0	1	1	21.6	3	True

[0103] Table 2, above, shows output from execution of the example code, which includes values for energy, occurrences and feasibility. As to feasibility, the presence of five edges makes an occurrence feasible; hence, occurrences with less than five edges (e.g., four edges) are not feasible. As to energy, the highlighted row with 35 occurrences is feasible (five edges) and has a lower energy than the other feasible occurrences of the next row. Thus, the solution corresponds to the entries of the highlighted row.

[0104] Contrary to classical algorithm, the extension of the foregoing quantum implementation to the constrained Delaunay case tends to be achievable. For example, to extend, the formulation can involve adding constrained edges to the input edges and updating the minimization constraints accordingly, which is demonstrated in more details in the example below.

[0105] Consider the same input as the example of Figure 3 with the edge 13 as a constrained edge. In this case, the minimization function is the same and the constraints become:

$q_0 + q_1 + q_2 + q_4 + q_5 = 5$, existing input edges

$\frac{\sum_{j \neq 0} w_{0j} \times q_0 q_j}{2} = 1$, constraint C1 for q_0

$\frac{\sum_{j \neq 1} w_{1j} \times q_1 q_j}{2} = 1$, constraint C1 for q_1

$\frac{\sum_{j \neq 2} w_{2j} \times q_2 q_j}{2} = 1$, constraint C1 for q_2

$\frac{\sum_{j \neq 5} w_{5j} \times q_5 q_j}{2} = 1$, constraint C1 for q_5

$\left(\frac{\sum_{j \neq 3} w_{3j} \times q_3 q_j}{2}\right) \%2 = 0$, constraint C2 for q_3

$\frac{\sum_{j \neq 4} w_{4j} \times q_4 q_j}{2} = 2$, constraint C2 for existing internal edge q_4

Notation is equal to 2 or 0 (%)

[0106] As explained, a framework enables implementation of one or more quantum computing techniques. Such a framework can facilitate finding the best solution from all feasible solutions, increasing reward while reducing costs and/or risks, solving problems for producing more energy with lesser cost, lesser emissions, improved efficiency, etc.

[0107] As explained combinatorial problems can be impractical to solve using classical computing techniques as the number of possible solutions can be very large. As mentioned, well scheduling can be a combinatorial problem where a solution space can be impractically large, for example, where the order in which production wells are drilled can impact feasibility, revenue, etc. (e.g., consider size of N! for N wells).

[0108] Table 3. Scaling of Well Scheduling

N	2	3	4	5	6	7	8	9	10	20	50
N!	2	6	24	120	720	5040	40320	362880	3628800	2.4e+18	3.0e+64

[0109] Using classical computing techniques for various problems can be prohibitive in that exhaustive searches may not be tractable such that approximation algorithms are required where solutions may be far from optimal.

[0110] As explained, a framework can solve geometrical optimization problems where, for example, one or more types of problems may be cast in terms of geometrical optimization using, for example, Delaunay triangulation.

[0111] As to various examples of quantum computing techniques, consider gate-based techniques and quantum annealing techniques. These techniques differ and underlying hardware can differ as well. For example, various quantum computers can be gate-based while others can be based on quantum annealing. As an example, a framework may utilize one or more of a gate-based approach and a quantum annealing approach to solve a combinatorial problem.

[0112] As an example, a one-bit representation can use a Dirac notation such as:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

[0113] In such an example, a tensor product can be represented as follows:

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} a\begin{pmatrix} c \\ d \end{pmatrix} \\ b\begin{pmatrix} c \\ d \end{pmatrix} \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$

[0114] In such an example, a quantum state of one qubit can be represented as follows:

$$|q\rangle = \binom{a}{b}, \text{ with } a^2 + b^2 = 1$$

$$a^2 \text{ probability of } 0$$

$$b^2 \text{ probability of } 1$$

[0115] In such an example, when measured, the quantum state of the qubit collapses to 0 or 1.

[0116] A fundamental principle of quantum mechanics adheres to: “any two or more quantum states can be added together, superposed, and result in another valid quantum state”.

[0117] A quantum computing technique may utilize a Hadamard gate-based approach, where mathematically it can be represented as follows:

$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}\binom{1}{0} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}\binom{0}{1} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

[0118] In the classical world, if two people read the same paper, they will have learned the same information. If a third person comes along and reads the same paper, that person too will have learned this information. Hence, all three people in this case are perfectly correlated, and they will remain correlated even if they are separated from each other.

[0119] In the quantum world, two people can read the same quantum paper, and yet we will not learn what information is actually contained in the paper until they get together and share their information. However, when they are together, they find that they can unlock more information from the paper than they initially thought possible. Thus, quantum entanglement goes much further than perfect correlation.

[0120] Figure 6 shows various example formulations 610, 620 and 630 as related to gate-based approaches to quantum computing. In Figure 6, a controlled-NOT gate (CNOT) 610 is shown, also known as a controlled-x (CX) gate, which acts on a pair of qubits, with one acting as a control and the other acting as a target. The CNOT gate performs a NOT on the target whenever the control is in state $|1\rangle$. If the control qubit is in a superposition, this gate creates entanglement.

[0121] All unitary circuits can be decomposed into single qubit gates and CNOT gates. Because the two-qubit CNOT gate costs much more time to execute on real hardware than single qubit gates, circuit cost is sometimes measured in the number of CNOT gates.

[0122] In Figure 6, a Hadamard gate 620 is also shown, denoted "H". The H, or Hadamard, gate rotates the states $|0\rangle$ and $|1\rangle$ to $|+\rangle$ and $|-\rangle$, respectively. It is useful for making superpositions. If there is a universal gate set on a classical computer and a Hadamard gate is added, it becomes a universal gate set on a quantum computer.

[0123] Also shown in Figure 6, is a representation 630 where, if the product state of two qubits cannot be factored then they are said to be entangled. In this example, when measured this product state collapses to:

50% at $|\mathbf{00}\rangle$,
50% at $|\mathbf{11}\rangle$
and not
25% at $|\mathbf{00}\rangle$,
25% at $|\mathbf{01}\rangle$,
25% at $|\mathbf{10}\rangle$,
25% at $|\mathbf{01}\rangle$

[0124] An H gate approach may be implemented using a platform such as, for example, the GOOGLE Quantum AI platform (Google, Inc., Mountain View, California), which includes various "cirq" methods. Cirq is a Python software library for writing, manipulating, and optimizing quantum circuits, and then running them on quantum computers and quantum simulators.

[0125] As explained, a computation can be performed by applying a sequence of gates to a system of qubits. In such an approach, there can be considerable control of qubits and their interactions, which may be leveraged to solve complex problems.

[0126] As explained, an adiabatic theorem approach may be implemented as a quantum computing technique (e.g., consider quantum annealing). For example, a quantum system can remain in its ground state if the Hamiltonian governing its dynamics changes slowly, and slowly depends on the smallest energy difference between the ground state and the first excited state exhibited during the evolution of the system (minimum gap).

[0127] In an adiabatic approach, a formulation can start with a relatively straightforward to solve initial Hamiltonian H_i and then evolve slowly to the final Hamiltonian H_f that corresponds to the optimization problem where the final ground state of the quantum system corresponds to the solution of the optimization problem:

$$H(t) = A(t)H_i(t) + B(t)H_f(t)$$
$$A(t) + B(t) = 1$$
$$A(t = t_0) = 1, \; B(t = t_0) = 0$$
$$A(t = t_f) = 0, \; B(t = t_f) = 1$$

[0128] As quantum systems tend to be quite sensitive to background noise and thermal fluctuation, a quantum system may leave its ground state due to its environment. In such a scenario, the strict adiabatic conditions are not guaranteed. For example, a problem may converge to a local optimum.

[0129] The quantum computing technique of quantum annealing is an approximation where the time to switch from H_i to H_f is estimated (because minimum gap is not known a priori). Such an approach can involve repeating an annealing process a number of times (e.g., according to an annealing schedule that may be of the order of microseconds or tens of microseconds, for example, consider 20 microseconds). As explained, an individual run may be quite short where a number of runs may be more than 10, more than 100, more than 1000, where a total solution time from a quantum computer may be less than one second. Quantum annealing tends to converge, with high probability, to the global optimum.

[0130] Quantum annealing can be described with respect to qubits where a qubit can be in a state of 0 or 1 where each qubit can be encoded in a circulating current with a corresponding magnetic field. Such a description finds parallels in nuclear magnetic resonance where a nuclear spin creates a magnetic field that, for example, may be aligned or not aligned with a static magnetic field. A qubit, as a quantum construct, can be in a superposition of a 0 state and a 1 state at a point in time. In a quantum annealing process, a qubit in superposition is driven to either the 0 state or the 1 state.

[0131] The underlying physics of a quantum annealing process can be described with respect to an energy diagram. For example, consider a parabola with a single valley or well that corresponds to the lowest energy state, which is a superposition state. In such an example, energy added as part of quantum annealing raises a barrier such that two wells exist, one for the 0 state and another for the 1 state, where the potential of each (e.g., energy of each) well is the same. However, the system can be disturbed through application of a bias, for example, in the magnetic analogy, by applying an external magnetic field that is aligned with the 1 state such that the energy of the 1 state well is lower than the energy of the 0 state well. In NMR, a parallel exists due to the presence of a strong static magnetic field where nuclear spins with magnetic fields aligned with that strong static magnetic field are in a lower energy state than those with magnetic fields anti-aligned with that strong static magnetic field. In terms of probabilities, the probability of a qubit ending up in the lower energy well is greater than the probability of the qubit ending up in the higher energy well. Hence, through use of a bias, the probabilities of a qubit being in a 0 state or in a 1 state can be controlled.

[0132] Further, various processes can be linked such that influence can exist between qubits. For example, a coupler can cause two qubits to be both 0 or both 1. Or, for example, a coupler can cause two qubits to be in opposing states. The use of a coupler can increase the number of possible states in a system with two qubits to four: 0,0, 0,1, 1,0 and 1,1. In such an example, the relative energy of each state depends on the bias applied to each qubit and the coupling between them. If a coupler wants two qubits to be the same, it lowers the energy of that coupled state; similarly, if a coupler wants two qubits to be opposite, it lowers the energy of that coupled state. As an example, a framework can provide for selection of biases and couplers. For example, a user may choose a set of biases and a set of couplings for a system of qubits to define an energy landscape where a quantum annealer finds a state in the energy landscape that is the lowest energy state of the system.

[0133] With two qubits, in quantum annealing, there can be four states, while with three qubits, the number of states is eight. Hence, the number of states increases exponentially with the number of qubits. As explained, a quantum annealing approach can commence with a number of unconnected qubits, each in a superposition state. In such an approach, biases and couplings can be introduced such that the qubits become entangled where the probability of an individual qubit of being in a 0 state or a 1 state is altered (e.g., from an initial 50/50 probability). Ultimately, the qubits reach their low energy states, which, for the system of qubits, is the lowest energy state. In the D-Wave platform, in general, the lowest energy state can be determined in a matter of milliseconds. For example, each run can take approximately 20 microseconds where a number of runs may be of the order of 1000 (or more) for a total of 20 milliseconds (e.g., generally less than one second). As explained, a problem can be cast as a quantum problem where a solution may be found, for example, in a very short period of time (e.g., under one second).

[0134] Figure 7 shows an example of a method 700 that includes a superposition state as a single valley or well that is split by a barrier into two valleys of equal probability. As explained, a bias can be applied such that the probabilities differ.

[0135] Figure 8 shows some examples of couplers 800. As explained, a quantum annealing approach to quantum computing can involve use of bias and/or coupling.

[0136] Figure 9 shows an example of a quantum computing environment 900 that includes various mapping methods, a sampler API, samplers, and compute resources, which can include CPUs, GPUs and QPUs. As an example, a quantum computing environment can include features for one or more gate-based approaches and/or features for one or more quantum annealing (e.g., adiabatic) approaches.

[0137] Figure 10 shows an example of a method 1000 and an example of a system 1090 that can be a computational framework for performing one or more actions of the method 1000. As shown in Figure 10, the method 1000 can include a formulation block 1010 for formulating an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space; a determination block 1020 for determining an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and a performance block 1030 for, using the optimal solution, performing a subsequent operation of the industrial workflow.

[0138] Figure 10 also shows various computer-readable media (CRM) blocks 1011, 1021 and 1031. Such blocks can include instructions that are executable by one or more processors, which can be one or more processors of a computational framework, a system, a computer, etc., which may be or include a quantum computer (e.g., QPUs, etc.). A computer-readable medium can be a computer-readable storage medium that is not a signal, not a carrier wave and that is non-transitory. For example, a computer-readable medium can be a physical memory component that can store information in a digital format.

[0139] In the example of Figure 10, a system 1090 includes one or more information storage devices 1091, one or more computers 1092, one or more networks 1095 and instructions 1096. As to the one or more computers 1092, each computer may include one or more processors (e.g., or processing cores) 1093 and memory 1094 for storing the instructions 1096, for example, executable by at least one of the one or more processors. In the example of Figure 10, the processing cores can include one or more QPUs (quantum processing cores). As an example, a computer may include one or more network interfaces (e.g., wired or wireless), one or more graphics cards, a display interface (e.g., wired or wireless), etc. The system 1090 can be specially configured to perform one or more portions of the method 1000 of Figure 10. As an example, instructions as in the blocks 1011, 1021 and 1031 may be included in the instructions 1096 as part of a framework such as, for example, a continuous source reflection seismology framework. Such a framework may be part of a larger framework that can include features for handling seismic survey data, generating images, generating models, etc.

[0140] Figure 11 shows an example of a framework 1100 that includes features for mesh generation, well sequence determinations, production optimization and one or more other types of industrial workflows that can involve optimizations. As shown, the framework 1100 can include features for quantum annealing and/or gate-based quantum computing. In the example of Figure 11, a development toolkit is included such as an SDK, etc., which may provide various features for integration of features of the framework 1100 with one or more other frameworks (see, e.g., Figure 1).

[0141] As an example, a method can include formulating an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space; determining an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and, using the optimal solution, performing a subsequent operation of the industrial workflow. As mentioned, a mesh generation can be an operation of an industrial workflow, which can include rendering a visualization using a mesh, performing a simulation using a mesh, etc., where, for example, results from visualization and/or simulation can be utilized to improve the industrial workflow. As explained, an optimization problem can pertain to placement of equipment, performance of field operations, etc. In the oil and gas industry, an industry workflow can involve identifying hydrocarbons in a subsurface geologic region, determining where to place a well, determining how to fracture a reservoir, determining how to perform an EOR process, drilling a well, performing an investigation using field equipment, injecting fluid, producing fluid, processing fluid, etc.

[0142] As explained, a problem can be formulated with constraints, which may be cast in geometric terms, optionally with geometric meaning. For example, a mesh problem may impose constraints as to one or more positions of nodes, one or more locations of edges, etc. As an example, consider a constraint that corresponds to a physical structure such as a geobody, a discontinuity (e.g., a fault, a fracture, etc.), a wellbore, etc. A problem can be solved to determine an optimal mesh in a region that includes a physical structure or physical structures. As explained, a quantum computing approach may be rapid such that a solution can be generated and used in a workflow to expedite the workflow.

[0143] As an example, a method can include formulating an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space; determining an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and, using the optimal solution, performing a subsequent operation of the industrial workflow. In such an example, the formulating can implement Delaunay triangulation in two or more dimensions.

[0144] As an example, determining an optimal solution can include determining optimal positions of nodes and/or edges. As an example, a method can include determining an optimal solution that includes determining an optimal sequence of nodes.

[0145] As an example, a method can include implementing quantum gates using a quantum computer and/or can include implementing quantum annealing using a quantum computer.

[0146] As an example, an operation of an industrial workflow can be a mesh generation operation where an optimal solution is for an optimal mesh. In such an example, a subsequent operation of the industrial workflow can include a simulation of physical phenomena using the optimal mesh.

[0147] As an example, an optimal solution can include a sequence for drilling wells where the sequence can be optimal subject to one or more constraints. In such an example, a subsequent operation can include drilling at least one of the wells.

[0148] As an example, a quantum computer can be implemented to determine an optimal solution in less than one second.

[0149] As an example, a method can include formulating that includes implementing at least one bias that adjusts probabilities for quantum states of at least one qubit and/or implementing at least one coupler that links quantum states of at least two qubits.

[0150] As an example, a problem can involve a number of nodes that is greater than 20. As an example, a number of nodes may determine to a number of qubits for generating an optimal solution using a quantum computer. As an example, a number of variables may scale to a power of a number of nodes (e.g., points).

[0151] As an example, an optimization problem can be a combinatorial optimization problem.

[0152] As an example, a method can include determining an optimal solution using minimizing energy or maximizing energy of a geometrical construct. In such an example, minimizing energy or maximizing energy can include using an energy landscape defined by energy states of qubits.

[0153] As an example, a method can include formulating a problem using a constrained Delaunay triangulation in multiple dimensions where determining an optimal solution to the problem includes using quantum annealing.

[0154] As an example, an edge can be a path from one node (e.g., point) to another node (e.g., point). For example, consider a well planning problem where nodes represent possible well positions and where edges can represent paths from one well position to another, which, for example, may be a physical path for which equipment can be transported from one position to the next (e.g., for drilling wells, etc.).

[0155] As an example, a system can include a processor; memory accessible by the processor; and processor-executable instructions stored in the memory that are executable to instruct the system to: formulate an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space; determine an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and, using the optimal solution, perform a subsequent operation of the industrial workflow.

[0156] As an example, one or more computer-readable storage media can include computer-executable instructions executable to instruct a computer to: formulate an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space; determine an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and, using the optimal solution, perform a subsequent operation of the industrial workflow.

[0157] A computer-readable storage medium (or computer-readable storage media) is non-transitory, not a signal and not a carrier wave. Rather, a computer-readable storage medium is a physical device that can be considered to be circuitry or hardware.

[0158] Figure 12 shows components of an example of a computing system 1200 and an example of a networked system 1210 with a network 1220. The system 1200 includes one or more processors 1202 (e.g., CPUs, GPUs, QPUs, etc.), memory and/or storage components 1204, one or more input and/or output devices 1206 and a bus 1208. In an example embodiment, instructions may be stored in one or more computer-readable media (e.g., memory/storage components 1204). Such instructions may be read by one or more processors (e.g., the processor(s) 1202) via a communication bus (e.g., the bus 1208), which may be wired or wireless. The one or more processors may execute such instructions to implement (wholly or in part) one or more attributes (e.g., as part of a method). A user may view output from and interact with a process via an I/O device (e.g., the device 1206). In an example embodiment, a computer-readable medium may be a storage component such as a physical memory storage device, for example, a chip, a chip on a package, a memory card, etc. (e.g., a computer-readable storage medium).

[0159] In an example embodiment, components may be distributed, such as in the network system 1210. The network system 1210 includes components 1222-1, 1222-2, 1222-3, . . . 1222-N. For example, the components 1222-1 may include the processor(s) 1202 while the component(s) 1222-3 may include memory accessible by the processor(s) 1202. Further, the component(s) 1222-2 may include an I/O device for display and optionally interaction with a method. The network may be or include the Internet, an intranet, a cellular network, a satellite network, etc.

[0160] As an example, a device may be a mobile device that includes one or more network interfaces for communication of information. For example, a mobile device may include a wireless network interface (e.g., operable via IEEE 802.11, ETSI GSM, BLUETOOTH, satellite, etc.). As an example, a mobile device may include components such as a main processor, memory, a display, display graphics circuitry (e.g., optionally including touch and gesture circuitry), a SIM slot, audio/video circuitry, motion processing circuitry (e.g., accelerometer, gyroscope), wireless LAN circuitry, smart card circuitry, transmitter circuitry, GPS circuitry, and a battery. As an example, a mobile device may be configured as a cell phone, a tablet, etc. As an example, a method may be implemented (e.g., wholly or in part) using a mobile device. As an example, a system may include one or more mobile devices.

[0161] As an example, a system may be a distributed environment, for example, a so-called “cloud” environment where various devices, components, etc. interact for purposes of data storage, communications, computing, etc. As an example, a device or a system may include one or more components for communication of information via one or more of the Internet (e.g., where communication occurs via one or more Internet protocols), a cellular network, a satellite network, etc. As an example, a method may be implemented in a distributed environment (e.g., wholly or in part as a cloud-based service).

[0162] As an example, information may be input from a display (e.g., consider a touchscreen), output to a display or both. As an example, information may be output to a projector, a laser device, a printer, etc. such that the information may be viewed. As an example, information may be output stereographically or holographically. As to a printer, consider a 2D or a 3D printer. As an example, a 3D printer may include one or more substances that can be output to construct a 3D object. For example, data may be provided to a 3D printer to construct a 3D representation of a subterranean formation. As an example, layers may be constructed in 3D (e.g., horizons, etc.), geobodies constructed in 3D, etc. As an example, holes, fractures, etc., may be constructed in 3D (e.g., as positive structures, as negative structures, etc.).

[0163] Although only a few example embodiments have been described in detail above, those skilled in the art will readily appreciate that many modifications are possible in the example embodiments. Accordingly, all such modifications are intended to be included within the scope of this disclosure as defined in the following claims. In the claims, means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents, but also equivalent structures. Thus, although a nail and a screw may not be structural equivalents in that a nail employs a cylindrical surface to secure wooden parts together, whereas a screw employs a helical surface, in the environment of fastening wooden parts, a nail and a screw may be equivalent structures.

CLAIMS

What is claimed is:

1. A method comprising:

formulating an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space;

determining an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and

using the optimal solution, performing a subsequent operation of the industrial workflow.

2. The method of claim 1, wherein the formulating implements Delaunay triangulation in two or more dimensions.

3. The method of claim 1, wherein the determining the optimal solution includes determining optimal positions of the nodes and/or optimal positions of the edges.

4. The method of claim 1, wherein the determining the optimal solution includes determining an optimal sequence of the nodes.

5. The method of claim 1, wherein the determining includes implementing quantum gates using the quantum computer.

6. The method of claim 1, wherein the determining includes implementing quantum annealing using the quantum computer.

7. The method of claim 1, wherein the operation is a mesh generation operation and wherein the optimal solution is for an optimal mesh.

8. The method of claim 7, wherein the subsequent operation includes a simulation of physical phenomena using the optimal mesh.

9. The method of claim 1, wherein the optimal solution includes a sequence for drilling wells.

10. The method of claim 9, wherein the subsequent operation includes drilling at least one of the wells.

11. The method of claim 1, wherein the quantum computer determines the optimal solution in less than one second.

12. The method of claim 1, wherein the formulating includes implementing at least one bias that adjusts probabilities for quantum states of at least one qubit.

13. The method of claim 1, wherein the formulating includes implementing at least one coupler that links quantum states of at least two qubits.

14. The method of claim 1, wherein the number of nodes is greater than 20.

15. The method of claim 1, wherein the optimization problem is a combinatorial optimization problem.

16. The method of claim 1, wherein the determining the optimal solution includes minimizing energy or maximizing energy of the geometrical construct.

17. The method of claim 16, wherein the minimizing energy or the maximizing energy includes using an energy landscape defined by energy states of qubits.

18. The method of claim 1, wherein the formulating includes using a constrained Delaunay triangulation in multiple dimensions and wherein the determining includes using quantum annealing.

19. A system comprising:

a processor;

memory accessible by the processor; and

processor-executable instructions stored in the memory that are executable to instruct the system to:

formulate an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space;

determine an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and

using the optimal solution, perform a subsequent operation of the industrial workflow.

20. One or more computer-readable storage media comprising computer-executable instructions executable to instruct a computer to:

formulate an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space;

determine an optimal solution for the optimization within the probabilistic solution space using a quantum computer; and

using the optimal solution, perform a subsequent operation of the industrial workflow.

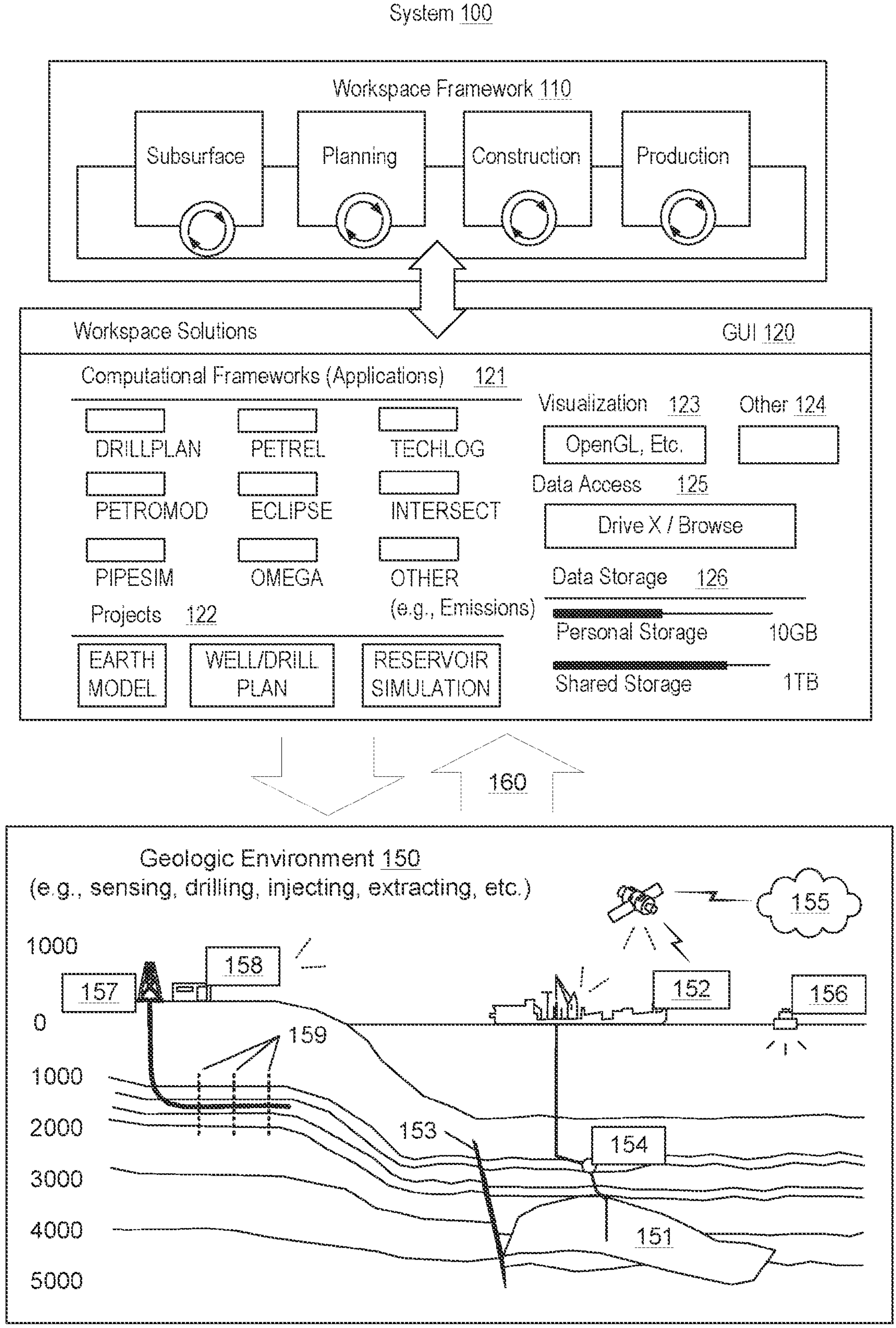

System 100
Workspace Framework 110
Subsurface
Planning
Construction
Production
Workspace Solutions
GUI 120
Computational Frameworks (Applications) 121
DRILLPLAN
PETREL
TECHLOG
PETROMOD
ECLIPSE
INTERSECT
PIPESIM
OMEGA
OTHER (e.g., Emissions)
Projects 122
EARTH MODEL
WELL/DRILL PLAN
RESERVOIR SIMULATION
Visualization 123
OpenGL, Etc.
Other 124
Data Access 125
Drive X / Browse
Data Storage 126
Personal Storage 10GB
Shared Storage 1TB
160
Geologic Environment 150
(e.g., sensing, drilling, injecting, extracting, etc.)
155
1000
158
157
152
156
0
159
1000
2000
153
154
3000
4000
151
5000

Fig. 1

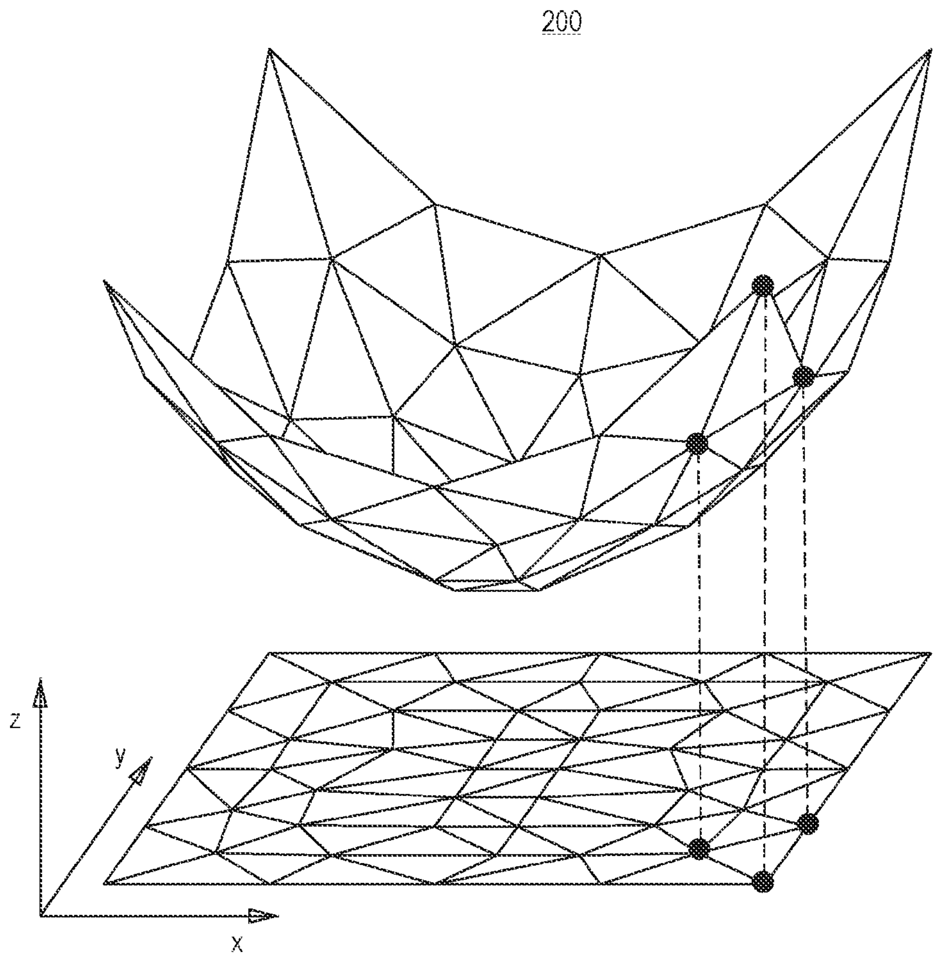


Fig. 2

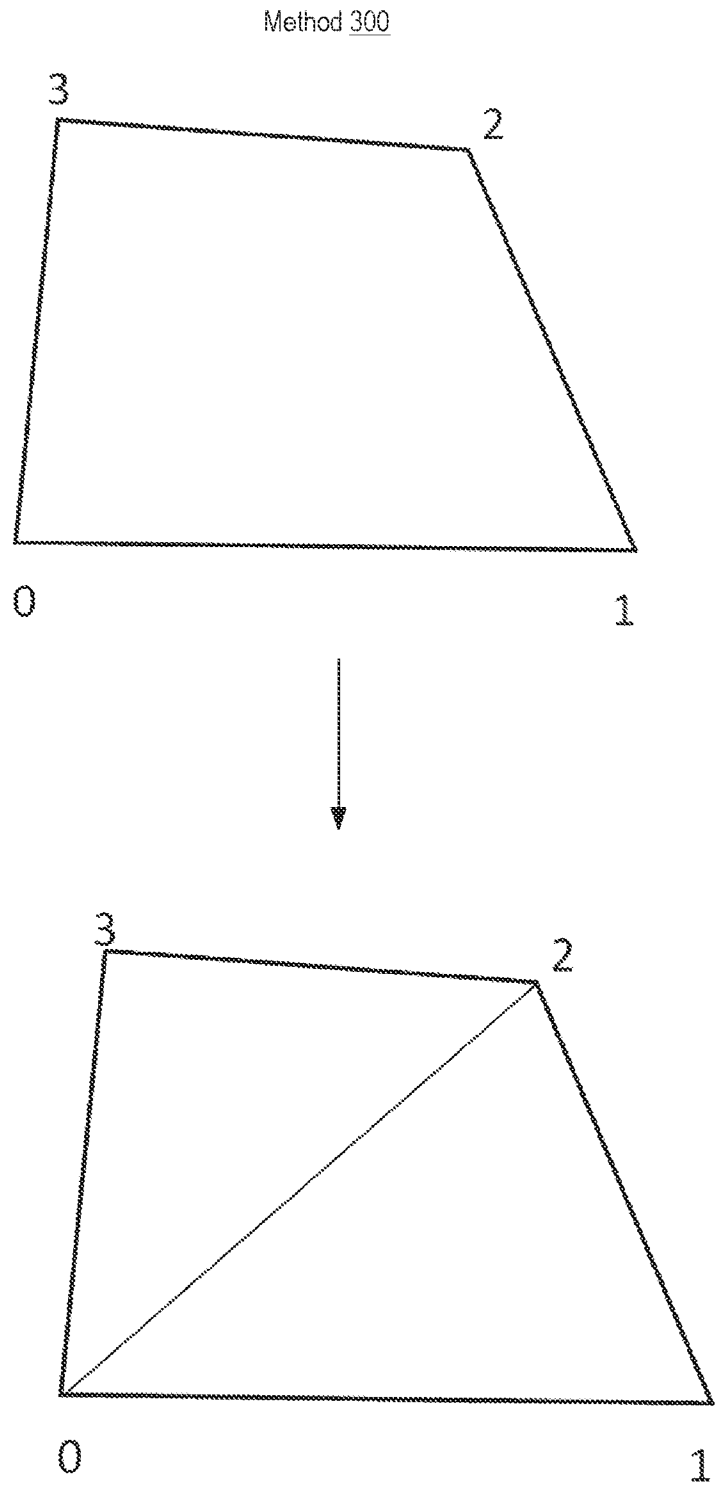


Fig. 3

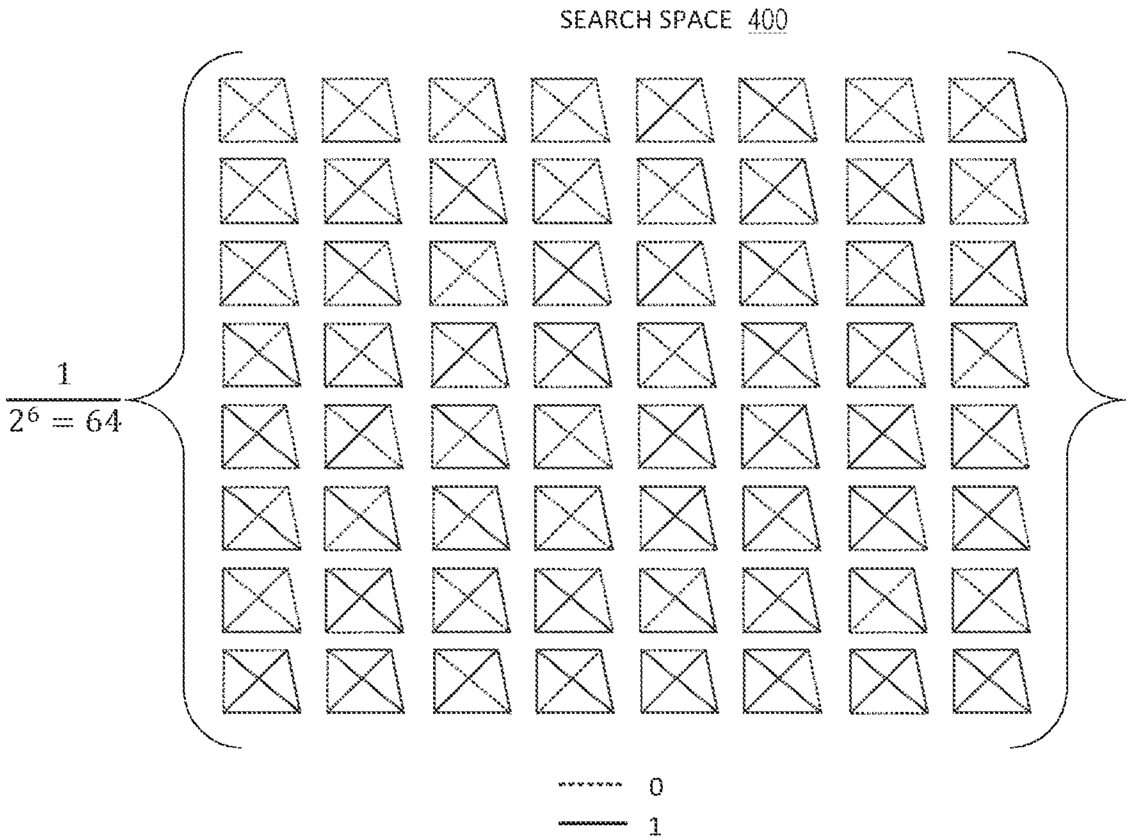
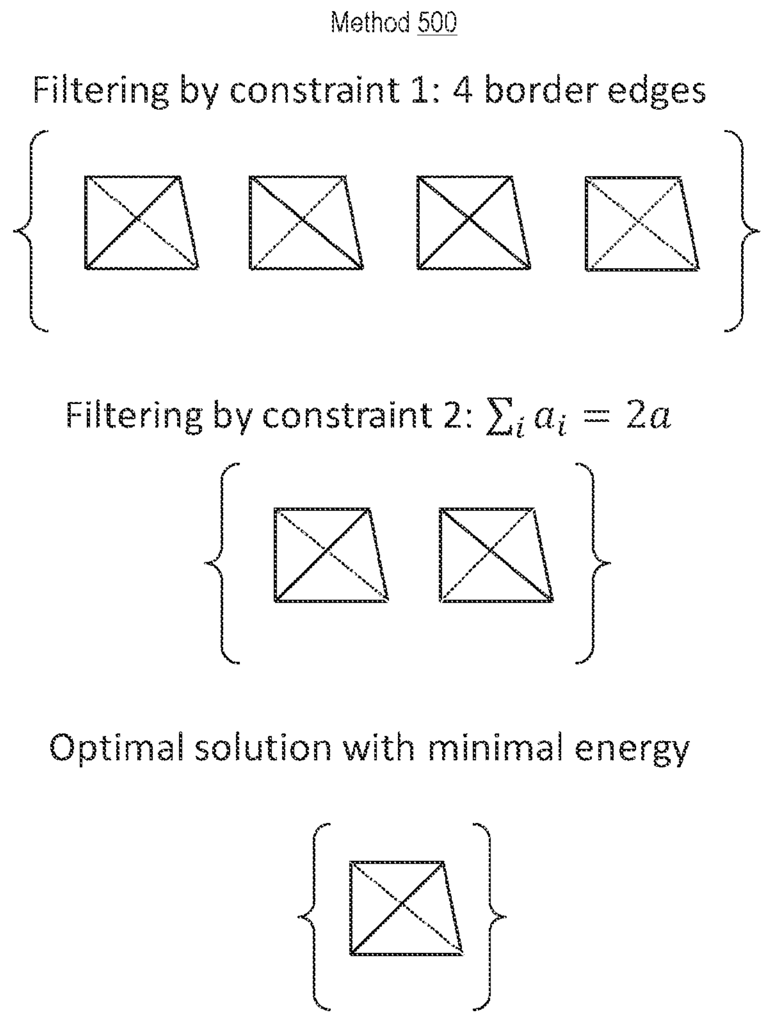


Fig. 4


Method 500
Filtering by constraint 1: 4 border edges
Filtering by constraint 2: $\sum_i a_i = 2a$
Optimal solution with minimal energy

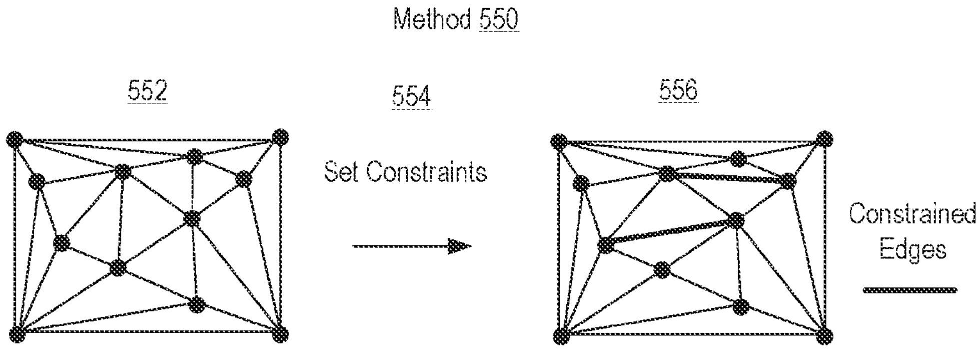
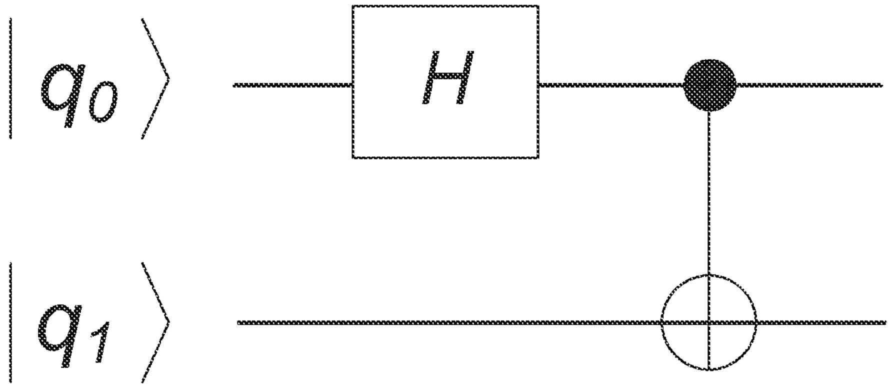

Method 550
552
554
556
Set Constraints
Constrained Edges

Fig. 5

610

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

620



630

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \text{ there is no } \begin{pmatrix} a \\ b \end{pmatrix} \text{ and } \begin{pmatrix} c \\ d \end{pmatrix} \text{ such that: } \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix}$$

Fig. 6

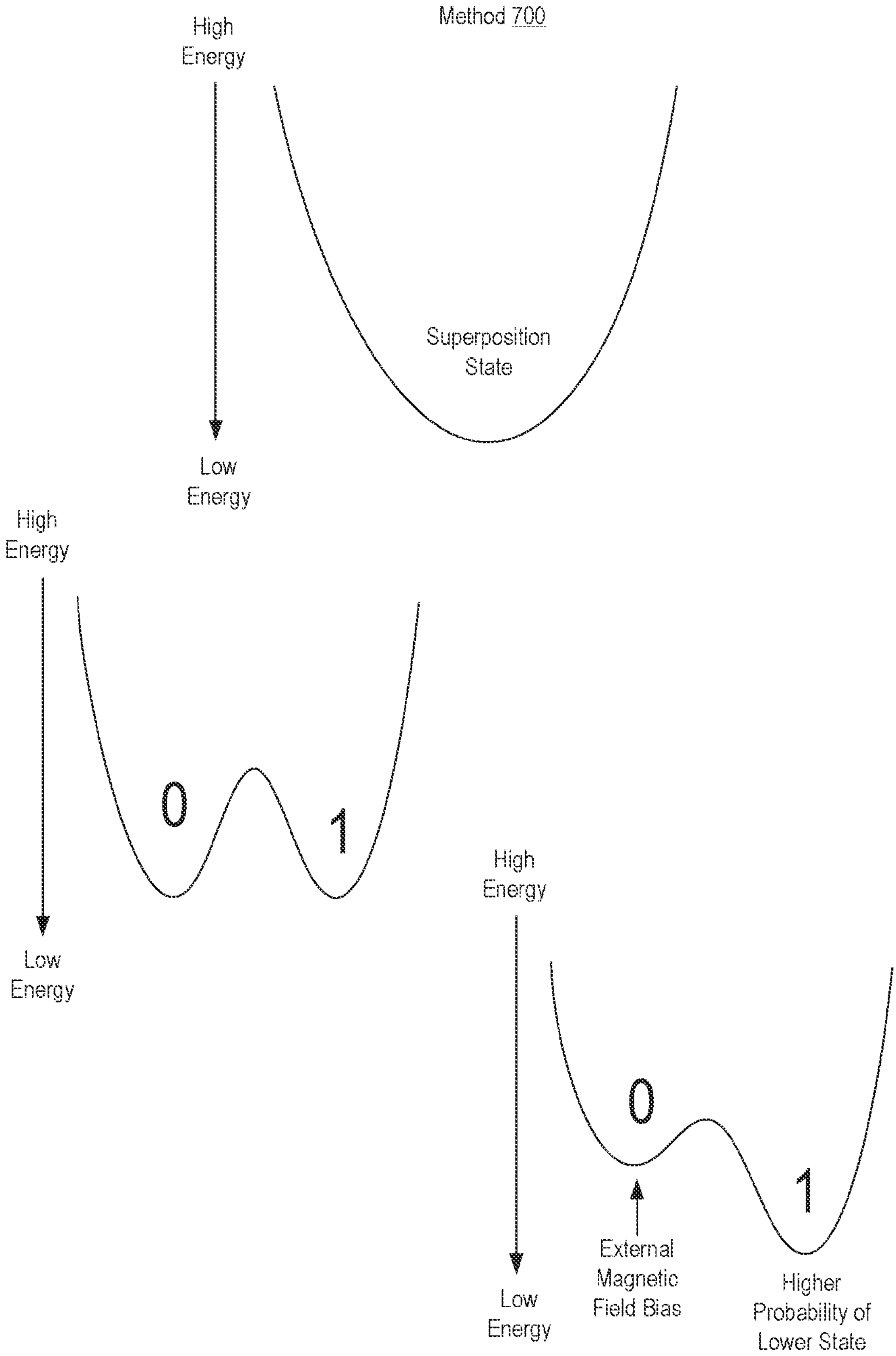

Method 700
High Energy
Low Energy
Superposition State
High Energy
0
1
Low Energy
High Energy
0
1
Low Energy
External Magnetic Field Bias
Higher Probability of Lower State

Fig. 7

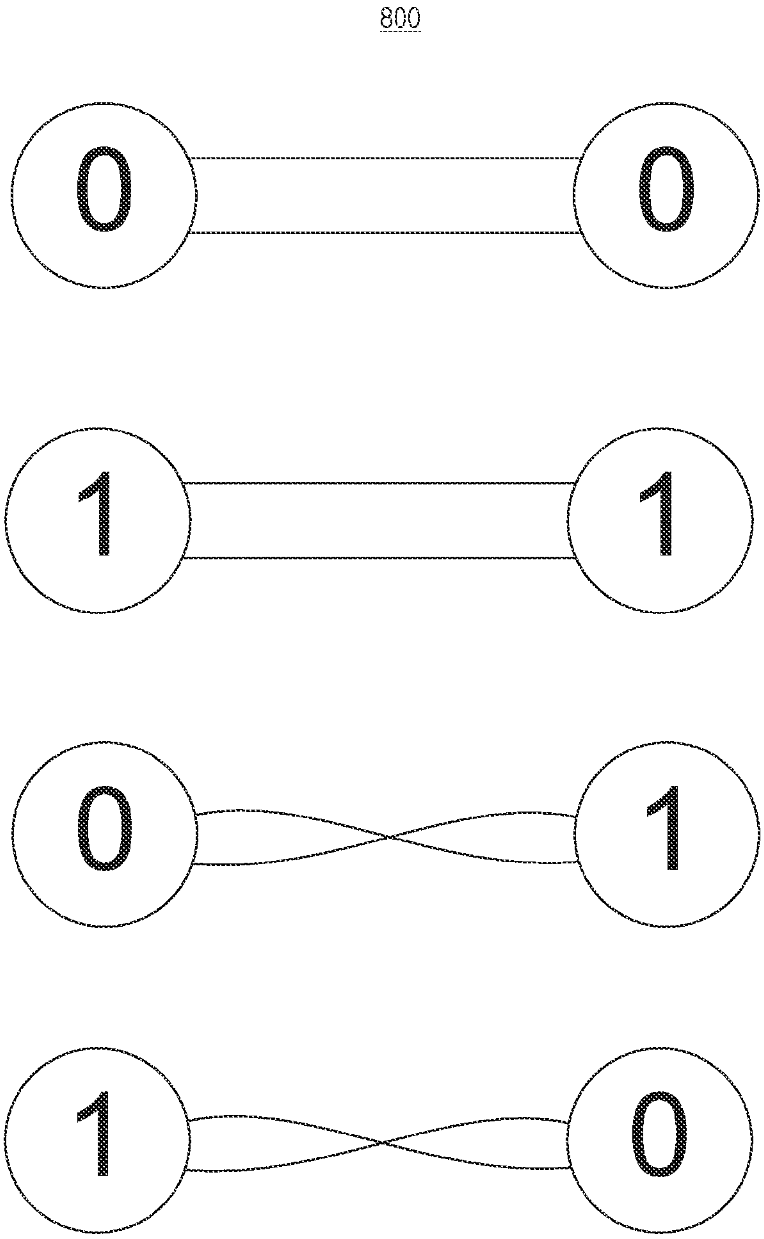


Fig. 8

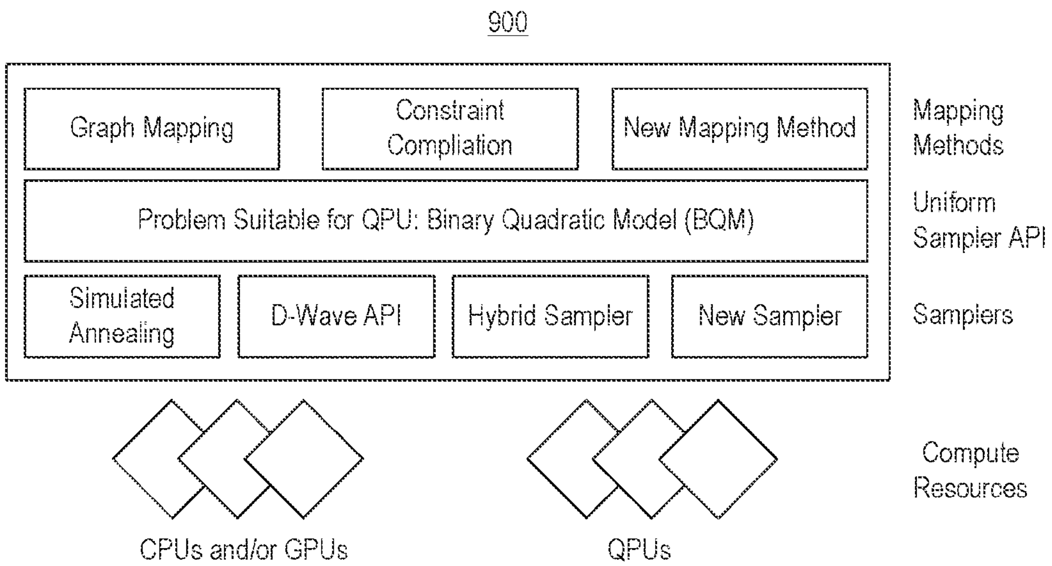

900
Graph Mapping
Constraint Compliation
New Mapping Method
Mapping Methods
Problem Suitable for QPU: Binary Quadratic Model (BQM)
Uniform Sampler API
Simulated Annealing
D-Wave API
Hybrid Sampler
New Sampler
Samplers
Compute Resources
CPUs and/or GPUs
QPUs

Fig. 9

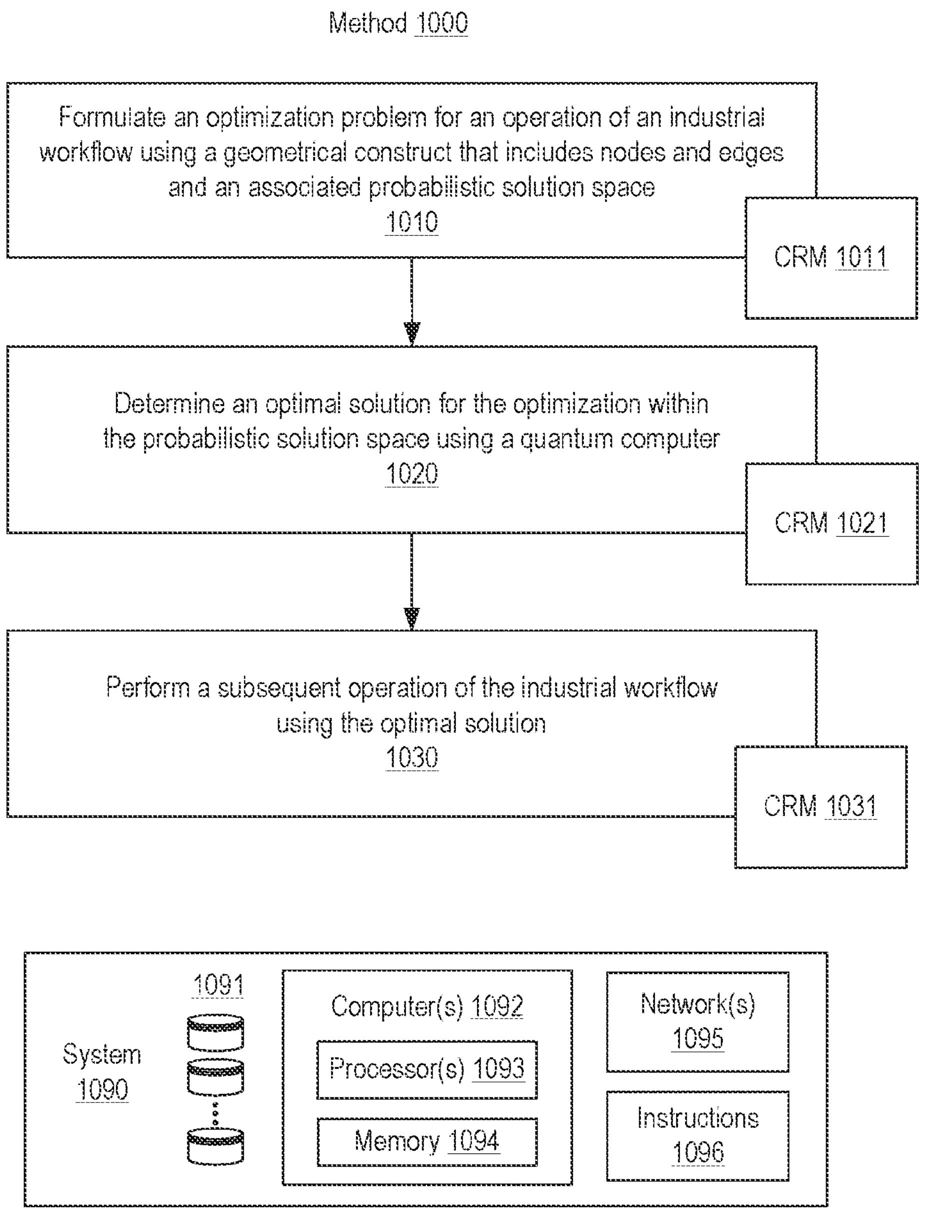

Method 1000
Formulate an optimization problem for an operation of an industrial workflow using a geometrical construct that includes nodes and edges and an associated probabilistic solution space
1010
CRM 1011
Determine an optimal solution for the optimization within the probabilistic solution space using a quantum computer
1020
CRM 1021
Perform a subsequent operation of the industrial workflow using the optimal solution
1030
CRM 1031
System
1090
1091
Computer(s) 1092
Processor(s) 1093
Memory 1094
Network(s)
1095
Instructions
1096

Fig. 10

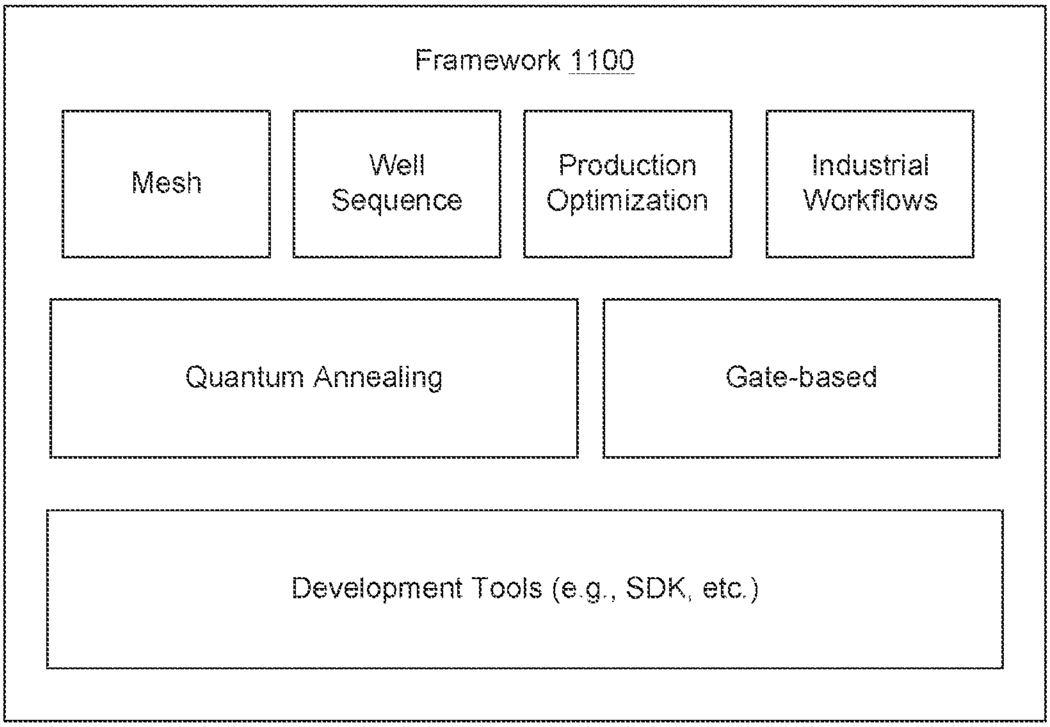

Framework 1100
Mesh
Well Sequence
Production Optimization
Industrial Workflows
Quantum Annealing
Gate-based
Development Tools (e.g., SDK, etc.)

Fig. 11

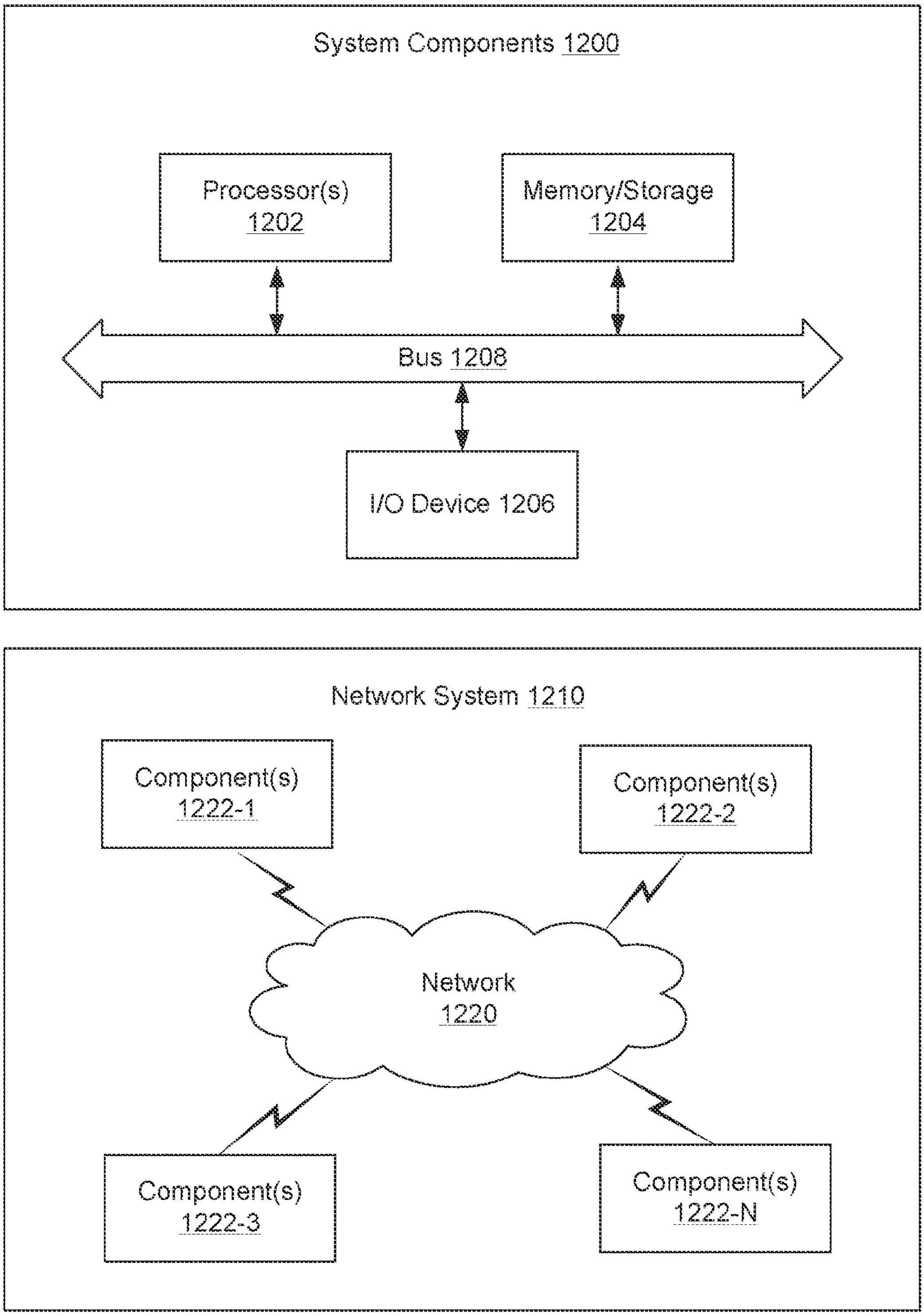

System Components 1200
Processor(s)
1202
Memory/Storage
1204
Bus 1208
I/O Device 1206
Network System 1210
Component(s)
1222-1
Component(s)
1222-2
Network
1220
Component(s)
1222-3
Component(s)
1222-N

Fig. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 24/16621

A. CLASSIFICATION OF SUBJECT MATTER

IPC - INV. G01V 1/28, G06F 30/20, G06N 10/00 (2024.01)

ADD. G01V 20/00 (2024.01)

CPC - INV. G01V 1/28, G06F 30/20, G06N 10/00

ADD. G01V 20/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012/0143575 A1 (Imhof et al.) 07 June 2012 (07.06.2012), entire document, especially abstract and para [0068], [0076], [0086], [0088]-[0090], [0098], Figs. 19A-19D, Figs. 20A-20D.	1-20
Y	US 2022/0114313 A1 (TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED) 14 April 2022 (14.04.2022), entire document, especially abstract and para [0035], [0046], [0053], [0176]-[0177].	1-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 April 2024 (19.04.2024)	MAY 30 2024

Name and mailing address of the ISA/US	Authorized officer
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Kari Rodriquez Telephone No. PCT Helpdesk: 571-272-4300