QUANTUM COMPUTING INC

Dirac-3 User Guide

May 2025 v0.0.4

Quantum Computing Inc
quantumcomputinginc.com
(703) 436-2161

Contents

Overview

Welcome to the Dirac-3 User Guide. Dirac-3 is the hybrid hardware following Entropy Quantum Computing (EQC), a physics-informed approach to solving complex, many-body, and highly interconnected multi-variable optimization problems. This manual provides instructions for the setup, installation, and usage as well as an overview of Dirac-3 hardware features.

For a better understanding of the fundamental physics behind QCi's EQC technology and operating principles of Dirac-3, please see "Entropy Computing: A Paradigm for Optimization in an Open Quantum System" [6].

Notice and Disclaimer

Any unauthorized use or reproduction of this User Guide may violate copyright laws and international treaties. The contents of this User Guide are provided for informational purposes only and may be subject to change without notice. Quantum Computing Inc. assumes no responsibility for errors or omissions in this document or for any damages, whether direct, indirect, incidental, consequential, or otherwise, arising from the use or inability to use the information contained herein.

All trademarks, trade names, logos, and service marks mentioned in this User Guide belong to their respective owners and are hereby acknowledged.

The use of this User Guide is limited to the intended purpose of assisting users in the proper installation, operation, and maintenance of the associated hardware and software. Any other use, including but not limited to commercial exploitation, distribution, or modification of the content, is strictly prohibited.

By accessing and using this User Guide, you acknowledge and agree to respect the copyright and intellectual property rights of Quantum Computing Inc. If you have any questions or requests regarding the use of this User Guide, please contact support@quantumcomputinginc.com

Warning

⚠**Non-Compliance Notice:** This device is undergoing the process to become compliant with CE or FCC emissions certification testings. Users should take appropriate precautions and use the device at their own risk.

Future Compliance Updates: QCi is committed to continuous improvement and will release updates or revisions to enhance compliance with standards and regulations. Users are advised to regularly check for updated documents on our website and follow any recommendations provided by Quantum Computing Inc.

Contact Information: All questions or concerns regarding the compliance status or safe use of this device should be directed to the QCi customer support team.

Document Revision History

Version	Date	Changes
0.0.1	February, 2024	N/A
0.0.2	March, 2024	• Section 8 - Updated programming example. • Section 9.2 - Added additional validation failure codes • Section 4.7 - Correct x and y axis labels on Figure 1
0.0.3	February, 2024	• Section 3 - Removed Software and OS requirements from specs
0.0.4	May, 2025	• Section 3 - Added integer solver content • Section 3 - Updated specs for connection • Section 3 - Added Dynamic Range and Solution Quality • Section 3 - Added Mean Photon Number and Shot Noise • Section 3 - Added Relaxation Schedule • Section 3 - Updated Computing Time • Section 3 - Updated Package Contents • Section 3 - Updated Setting up Dirac-3 with more details on rack-mounted installation, safety precautions, and example setup

1 A Specialized Quantum Hardware

Engineers and scientists at QCi have spent over a decade innovating elements, components, and designs for quantum photonic devices and developing a new approach for encoding and processing quantum information. This approach, from its inception, is aimed at developing energy-efficient, robust, scalable, and affordable devices for solving a large variety of real-world problems. The methodology applied in Dirac-3 is "Entropy Quantum Computing" (EQC).

1.1 Room Temperature

Many quantum technologies which use massive particles (such as electrons, ions, atoms, and superconductors) operate as closed quantum systems. This is necessary in order to isolate from environmental interference and preserve fragile quantum states from decoherence. Therefore, these systems require cryogenic cooling and elecromagnetic shielding from ambient conditions in order for them to function. In stark contrast to this, EQC is designed to operate as an open quantum system. In the EQC approach the coupling to the environment is critical to performing the desired computations.

1.2 SWAP-C

EQC uses photon number states in quantized optical time-frequency modes as the basis for quantum information processing. The bosonic nature of photons affords our system great stability in the operational environment and obviates the need for cooling or shielding. This substantially simplifies the system architecture while dramatically reducing the device Size, Weight, Power, and Cost (SWaP-C) of the device.

1.3 Decoherence

Loss or decoherence of a quantum state occurs through its coupling to an entropy source of many degrees of freedom. The apparent diminishing of quantum characteristics as a result is just a statistically averaged manifestation of many possible outcomes of this coupling. Second, vacuum is never quiet, although it does not appear to contain any energy or particles. In fact, there are enormous amounts of random fluctuations occurring at all times along each degree of freedom (DoF, also called modes) of the vacuum.

EQC is built upon these intriguing principles. Rather than trying to create and manipulate pristine qubits isolated from the environment, EQC embraces loss and decoherence, and turns entropy into fuel for its computing engine. In sharp contrast to any existing quantum computing platforms, there is no cryogenic cooling, and the systems can be made small, compact, and low-cost, just like a regular PC.

2 Motivation for Building Dirac-3

2.1 Limitations of Digital Classical Computing for Discrete Optimization

Discrete optimization problems involve identifying the most favorable solution from a finite and discrete set of possibilities. Some prominent examples include the traveling salesman problem, knapsack problem, graph coloring, spanning tree, matching, set covering, and set packing [7].

These problems are applicable to a wide array of fields, including scheduling, logistics, network design, drug discovery, and data analysis. Many discrete optimization problems are NP-hard, that is, no known algorithm can efficiently solve all instances of the problem in polynomial time relative to the input size. Consequently, as the problem size increases, the time required to find an exact optimal solution may grow exponentially. Advancements in hardware are necessary to meet these increasing computing demands. However, the saturation of the physical boundaries of semiconductor miniaturization poses a significant obstacle to further reducing transistor sizes. Simple parallel processing is not power efficient and does not meet the exponentially growing demand in data processing speed and capacity. Therefore, a logical progression is shifting away from universal Turing Machines and exploring alternative computing methodologies for specific tasks.

2.2 Alternative Computing

Recently, significant progress has been made in alternative computing approaches where the mathematical formulation can be mapped into the evolution of physical systems. This is the main motivation behind the proposal of quantum computers by Richard Feynman, a founder of the entire quantum computing field. These analog based computing methods that harvest solutions from natural phenomena include reservoir computing, neural network computing, quantum annealing, thermal relaxation, and coherent Ising machines [9], [4], [3]. These specialized hardware hold the promise of significantly reducing computation times and energy consumption, enabling the tackling of even larger and more complex problems. A significant portion of quantum computing research, particularly in the area of quantum annealing, focuses on solving Ising problems due to their direct applicability to combinatorial optimization problems. In fact, a good number of NP-complete combinatorial problems can be mapped to the binary Ising spin model. However, there are also many NP problems that do not naturally or directly map to the Ising model's framework of binary spin states, and developing an effective mapping that accurately represents the original problem within the Ising framework can be non-trivial. Furthermore, incorporating constraints that often accompany NP-hard problems into the Ising model can significantly increase the complexity of the problem formulation. This usually requires additional spins (qubits in the context of quantum computing) and carefully designing interaction terms to ensure that the constraints are properly enforced, which puts further constraints on its computing performance and capabilities.

2.3 Dirac-3

Dirac-3 on the other hand is built on a cost function that is fundamentally different from the Ising Hamiltonian. While a typical Ising Hamiltonian contains quadratic interaction terms over binary qubits, Dirac-3 supports higher-order interactions among high-dimensional quantum states (aka. qudits) encoded as photon-number states over multiple time-frequency modes. As such, Dirac-3 offers two significant advantages over a typical Ising solver:

- Dirac-3 can naturally represent non-binary discrete optimization problems
- Dirac-3 involves k-body interaction terms (k=1, 2, 3, 4, …) as opposed to the Ising machine that is limited to first and second order terms

Accordingly, Dirac-3 offers great potential in efficiently solving real-life discrete optimization problems as well as problems that naturally involve higher-order interaction terms. It also eliminates the otherwise additional complex encoding steps or the incorporation of auxiliary variables that generally add to the size of the problem in the case of an Ising solver.

3 Dirac-3 at a Glance

Solver Type	Constrained & Unconstrained Discrete Number Optimization
Objective Function	$E=\sum_{i=1}^{N} C_i V_i + \sum_{i,j=1}^{N,N} J_{ij} V_i V_j + \sum_{i,j,k=1}^{N,N,N} T_{ijk} V_i V_j V_k + \sum_{i,j,k,l=1}^{N,N,N,N} Q_{ijkl} V_i V_j V_k V_l + \sum_{i,j,k,l,m=1}^{N,N,N,N,N} P_{ijklm} V_i V_j V_k V_l V_m$
Hardware Type	Hybrid Analog Machine with Quantum Optics and Digital Electronics
CAPABILITY	
Maximum Number of Variables	949 (See Table (1) for details)
Order of Correlation	Any types of first- through fifth-order correlations, where the interaction amongst variables can be repulsive (positive correlation) or attractive (negative correlation).
Connectivity	All-to-all
System Power Consumption	under 100 Watts
ENVIRONMENTAL	
Storage Temperature	-25°C to 85°C
Operating Temperature	20°C to 27°C
Maximum Rate of Change	2°C per hour
OPERATION REQUIREMENTS	
Software Requirement	eqc-direct software package, Python 3.10.6 (recommended)
OS Requirement	Linux (recommended)
PHYSICAL CHARACTERISTICS	
Dimension	5U rack-mounted
Connection	1x 100/1000 Base-X Gigabit Ethernet (Optical) and MM 50/125 μ m Core, 850 nm, SC/PC

Order of Correlation	Maximum Number of Variables (Quasi-continuous)
First	949
Second	949
Third	135
Fourth	39
Fifth	19

Table 1: N-body interaction: Order of Correlation vs. Maximum Number of Variables assuming all-to-all connectivity

4 Solving Problems on Dirac-3

Dirac-3 solves problems involving the minimization or maximization of objective functions over discrete spaces by finding the ground state of a complex system with many inter-correlated variables. These variables correspond to the expected return of the following objective function:

$$E = \sum_{i=1}^{N} C_i V_i + \sum_{i,j=1}^{N,N} J_{ij} V_i V_j + \sum_{i,j,k=1}^{N,N,N} T_{ijk} V_i V_j V_k + \sum_{i,j,k,l=1}^{N,N,N,N} Q_{ijkl} V_i V_j V_k V_l + \sum_{i,j,k,l,m=1}^{N,N,N,N,N} P_{ijklm} V_i V_j V_k V_l V_m \quad (1)$$

Here, V_i denotes the value of each variable. C_i represents the linear contribution (or return) of each variable, and must be a real number. J_{ij}, T_{ijk}, Q_{ijkl}, and P_{ijklm} are higher-order interaction terms, also required to be real numbers.

Dirac-3 supports direct submission for minimization problems only. To handle maximization problems, users must perform a simple transformation by multiplying the objective function by -1 before submission.

4.1 Continuous Solver

Using amplitude encoding on each photon time-bin, Dirac-3 functions as a quasi-continuous optimization solver, subject to the constraint of a fixed resource R:

$$R = \sum_{i=1}^{N} V_i, R \in [1, 10000]$$

This means the resulting state vector V_i is continuous, with resolution determined by the system. Refer to section 4.4 for more details. Below are some examples. We evaluated the following quadratic cost function using Dirac-3:

$$E = 5x_1^2 + 4x_2^2 + 3x_3^2 + 2x_4^2 + x_5^2 \quad (2)$$

with the sum constraint R = 100; a typical raw output from Dirac-3 is

$$\begin{aligned} \text{Energy} &= 4379.86 \\ V_i &= [8.53924, 10.9957, 14.7423, 21.9537, 43.7687]; \end{aligned} \quad (3)$$

When the same Hamiltonian is optimized with a larger constraint R = 1000, the solution remains functionally similar, but the precision is reduced due to the fixed number of resolution levels:

$$\begin{aligned} \text{Energy} &= 437956 \\ V_i &= [87.5832, 109.487, 145.987, 218.989, 437.953]; \end{aligned} \quad (4)$$

This behavior reflects a fundamental tradeoff: increasing the solution space without increasing resolution granularity reduces effective precision.

4.2 Integer Solver

Using time-bin location encoding, Dirac-3 can also act as an unconstrained integer optimization solver. In this mode, the resulting state vector V_i consists of integers:

$$V_i \in \mathbb{N}, \quad 0 \leq V_i \leq 16$$

Users must specify the number of discrete states (levels) for each variable. For problems requiring more than 16 states, the continuous solver is recommended.

This flexible encoding allows Dirac-3 to support Quadratic Unconstrained Binary Optimization (QUBO) by setting the number of levels for each variable to 2. Furthermore, Dirac-3 supports mixed encoding, enabling some variables to have binary values (0 or 1) and others to have higher integer levels than 2.

4.3 All-to-All Connectivity

Photonic computing systems offer advantages in high connectivity compared to matter-based systems. Dirac-3, for instance, enables all-to-all interaction among variables. Users can input their Hamiltonian directly into Dirac-3, without the need to convert their problems to match the computing topology. This streamlines the process and results in faster time-to-solution, particularly as the number of connections increases quickly with problem size.

4.4 Dynamic Range and Solution Quality

Dynamic range of C_i, J_{ij}, T_{ijk}, Q_{ijkl}, P_{ijklm} are constrained to finite resolutions from optical modulator, digital-to-analog converter (DAC), input/output signal conditioning, and variation of temperature as well as dark count from single photon detector.

The analog nature of Dirac-3 causes the system's effective dynamic range to vary due to temperature fluctuations and device-level imperfections when measuring quantum states. While Dirac-3 can accept any signal within the digital system's readout range (up to $\approx 70dB$), it struggles to distinguish very small coupling terms due to its effective analog resolution limit, which is approximately 200:1, or 23 dB.

This limitation can be mathematically expressed as:

$$\frac{max(C_i, J_{ij}, T_{ijk}, Q_{ijkl}, P_{ijklm})}{min(C_i, J_{ij}, T_{ijk}, Q_{ijkl}, P_{ijklm})} \leq 200 \tag{5}$$

Furthermore, in order to resolve two distinct coupling terms, their difference must exceed the minimum resolvable level (i.e., the maximum term divided by the dynamic range). This constraint is written as:

$$|x - y| \geq \frac{max(C_i, J_{ij}, T_{ijk}, Q_{ijkl}, P_{ijklm})}{200} \tag{6}$$

$$x, y \in \{C_i, J_{ij}, T_{ijk}, Q_{ijkl}, P_{ijklm}\}, i \neq j \neq k \neq l \neq m \tag{7}$$

4.5 Mean Photon Number and Shot Noise

Coherent states quantum state of light closely resemble classical oscillation of light field, except from superimposed with quantum noise, specifically shot noise. Dirac-3 quantum optimization machine uses single photon count of each time-bin as a feedback signal. Therefore, the system experience Poisson noise taken advantage as a source of "entropy" or "fluctuation", constantly being boosted into each feedback loop. This parameter is defined as "quantum fluctuation coefficient" (QFC) that equals to $\frac{1}{\sqrt{N}}$, following the standard deviation of Poisson distribution where N the number of photon collected in each time-bin. Therefore, throughout the measurement and feedback process, the injection of quantum fluctuations enables a search in a near-infinite search space, which assists in bypassing the trapping of the quantum state into local minima. Users can specify QFC by integer from 1 to 100 where higher fluctuation corresponds to lower integer value.

While our entropy computing approach implementation is designed for photon number states or Fock states, it is practically difficult with the current state-of-the-art in deterministic photon number generation [8, 5]. In current hybrid implementation of Dirac-3, we use a common approximation of Fock states which is coherent states with ultra-low mean photon number μ[2] (MPN). Dirac-3 is maintained such that about 1% or less of pulses has more than 1 photon. Users have option to modify MPN to search for best parameter for their problems between the range from $6.67 * 10^{-5}$ to $6.67 * 10^{-3}$.

Example 3

We evaluate a non-convex quadratic optimization problem (QPLIB_0018) with 50 continuous variables over a fully connected weighted graph, selected from QPLIB, a library of quadratic programming instances [1]. The cost function of this problem is of the form $f(\mathbf{x}) = C^T\mathbf{x} + \mathbf{x}^T J\mathbf{x}$, where, $\mathbf{x} = (x_1, x_2, \cdots, x_{50})^T$ and $x_i \geq 0$.

Dirac-3 minimizes the energy for a polynomial solution $\mathbf{x}$ subject to following constraints:

$$\sum_i x_i = C, \quad x_i \geq 0$$

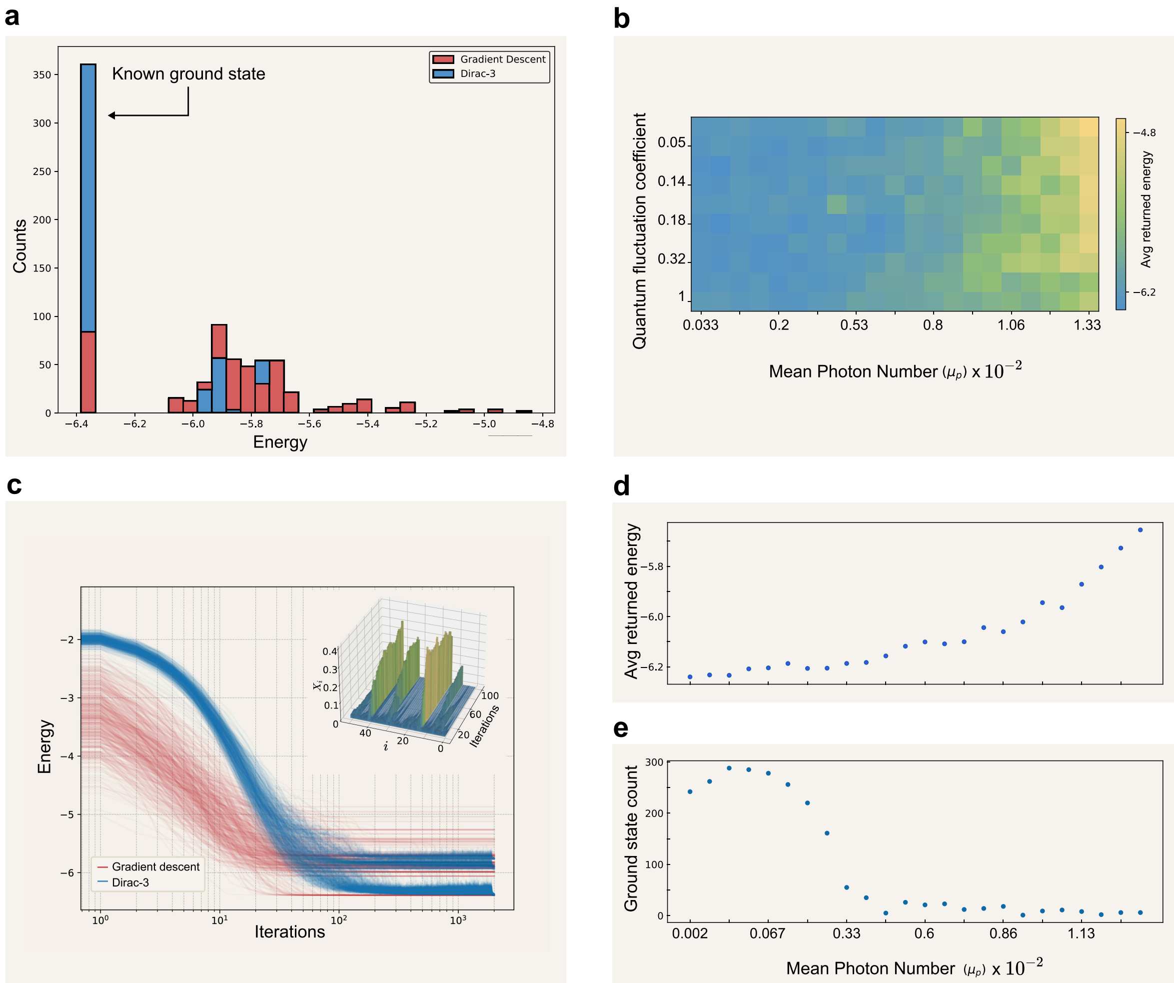


Figure 1: Solving a non-convex continuous optimization problem. A non-convex quadratic optimization problem with 50 variables is considered (QPLIB_0018). (a) Energy distribution over 500 runs of Dirac-3 (red) and gradient descent algorithm (blue). (c) Energy evolution versus the number of iterations on Dirac-3 (red) and gradient descent (blue). The inset shows the evolution of the solution versus iterations. (b) Investigate the relationship between the mean photon number (μ), quantum fluctuation coefficient,and key performance metrics - average returned energy. The transitioning from left to right, represents an increase in mean photon number (μ), showing the probability that Dirac-3 is operating in single-photon regime. Quantum fluctuation or shot noise is used as $\frac{1}{\sqrt{N}}$ where N is the number of photon count accumulated in each time-bin. From bottom to top of the vertical axis, this coefficient is gradually decreased. (d), (e) Average returned energy and number of ground states found after 500 runs are collected when mean photon number is decreased further up to 0.002 %, close to the level of dark count of single photon detector.

4.6 Relaxation Schedules

Dirac-3 offers four relaxation schedules, each determining the time allocated for the system to evolve and converge towards the ground state.

Continuous Solver

Relaxation Schedule	Mean Photon Number	Quantum Fluctuation Coefficient	Evolution Time
1	$3.33 * 10^{-4}$	1	t_1
2	$3.33 * 10^{-4}$	1	t_2
3	$3.33 * 10^{-4}$	1	t_3
4	$3.33 * 10^{-4}$	1	t_4

Integer Solver

Relaxation Schedule	Mean Photon Number	Quantum Fluctuation Coefficient	Evolution Time
1	$3.33 * 10^{-4}$	15	t_1
2	$3.33 * 10^{-4}$	15	t_2
3	$3.33 * 10^{-4}$	33	t_3
4	$3.33 * 10^{-4}$	15	t_4

The MPN parameter can be adjusted within the range of $[6.67 * 10^{-5}, 6.67 * 10^{-3}]$ and the QFC parameter can be adjusted within the range of [1,100]. Their default values are provided in Tables 4.6 and 4.6. Evolution time increases from t_1 (shortest) to t_4 (longest).

4.7 Analog Nature

An analog computing system maps a mathematical problem into physical phenomena and therefore it is a natural fit optimization purposes. However, that comes with a device dependency, due to both inevitable errors of the physical hardware and the inherent non-deterministic nature of the measurement outcomes. Therefore, it is not unusual that even for a simple problem that can be solved in digital computer, Dirac-3 will have some probability of returning a sub-optimal answer. For example, the problem:

$$E = -10x_1 - 10x_3 - 4x_2^2 - 5x_1x_3 \tag{8}$$

had the following energy distribution in Figure 2 when running on Dirac-3 with a sum constraint of 6 after 100 trials.

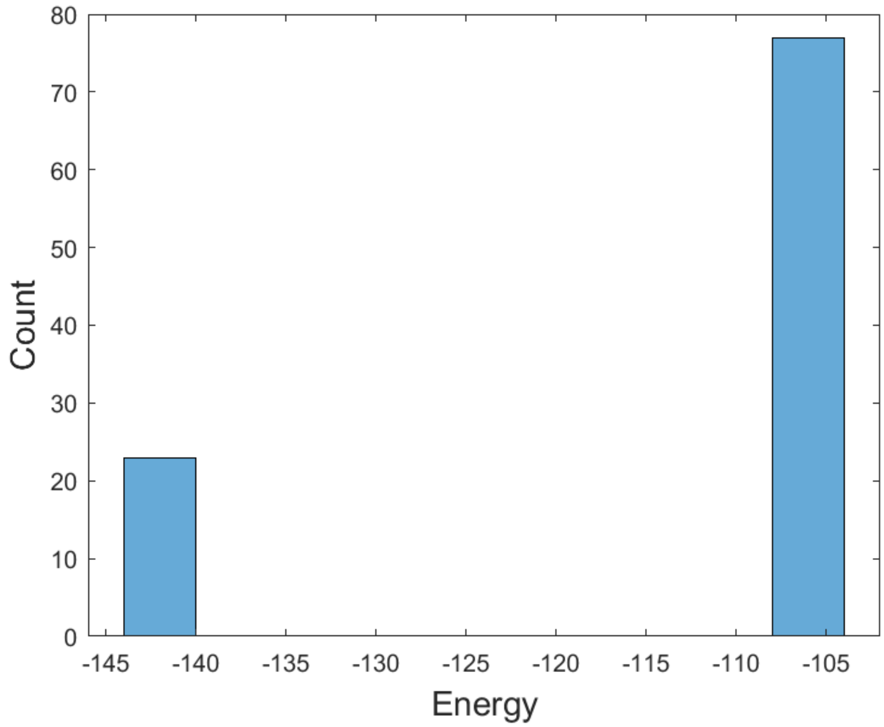


Figure 2: Example of a typical energy distribution results of objective function 8. Dirac-3 returns optimal solution 23% and sub-optimal 77% after 100 trials.

In Figure 3, a non-convex optimization on Dirac-3 is compared to classical gradient descent. Due to the non-convex nature of the problem, it is expected that the gradient descent has some probability of getting to global optimum and some probability of falling into a local optimum, depending on the starting point of the search. Taking advantage of quantum fluctuation, Dirac-3 is expected to be able to jump out of local minimum valleys with high probability. A non-convex problem from the Library of Quadratic Programming Instances QPLIB_0018 was modified with an offset equal to 1 in the linear terms and run on Dirac-3 50 times with schedule 2 and schedule 3.

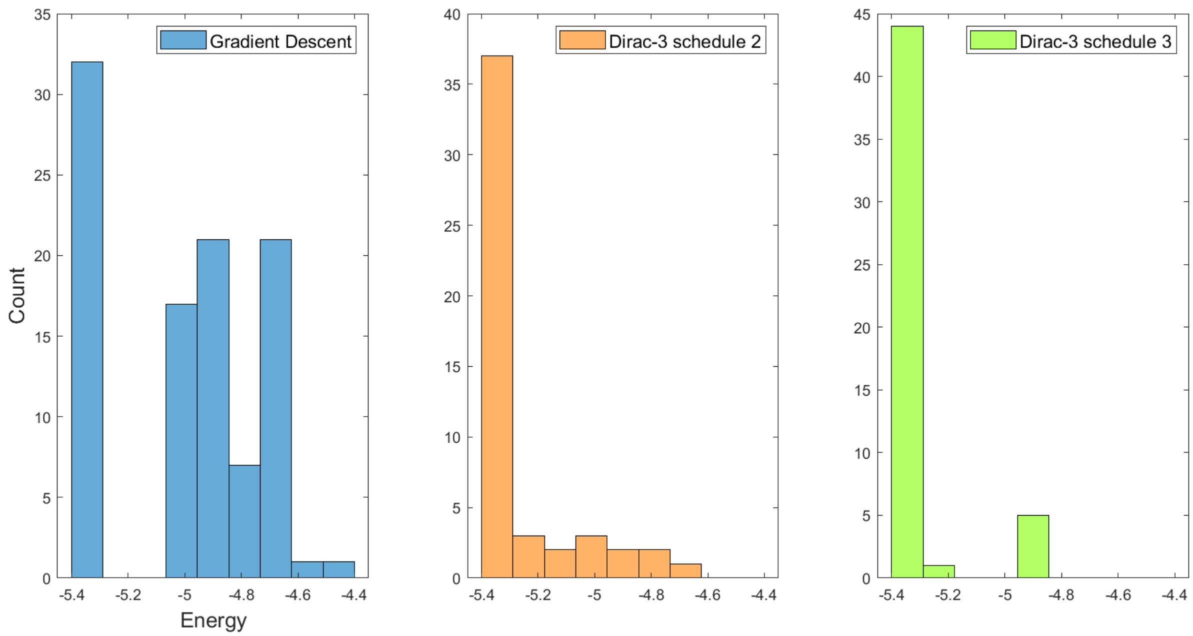


Figure 3: A typical distribution of energy over 50 runs. Dirac-3 performs superior to Classical Gradient Descent, returning optimal answer with high probability.

5 Understanding Dirac-3 Performance

5.1 Computing Time

Dirac-3 leverages digital electronics through the utilization of a field-programmable gate array (FPGA) to process quantum wavefunction measurements information. Matrix multiplication is used to emulate the global interaction of each “qudit,” with results feeding back into the quantum system and undergoing multiple iterations sequentially. Consequently, the overall computing time is the cumulative sum of each iteration.

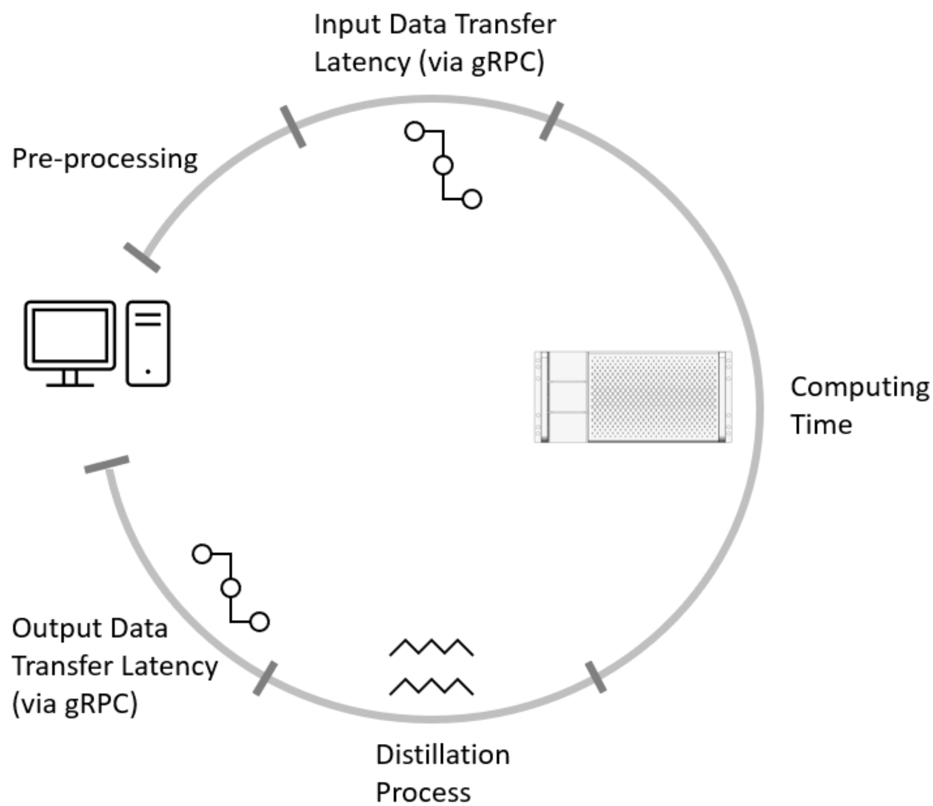


Figure 4: Computing time breakdown in Dirac-3 when it is connected to a single host computer

Figure 4 illustrates the breakdown of the total time users experience when running a problem with a host server/computer connected to Dirac-3.

Pre-processing: input validation and formatting.
Data Transfer Latency: data transfer in and out via gRPC
Computing Time: single photon counting and processing, matrix multiplication via FPGA, and optical control

6 Unpacking Dirac-3

Follow these steps:

1. Unpack and remove the device and the accessory kit from the shipping box.
2. Return the packing material to the shipping container, and save it for future use.
3. Verify that you have received the items shown in the “Package Contents” section. If any item is missing or damaged, contact your QCi Support representative or reseller for instructions.

6.1 Package Contents

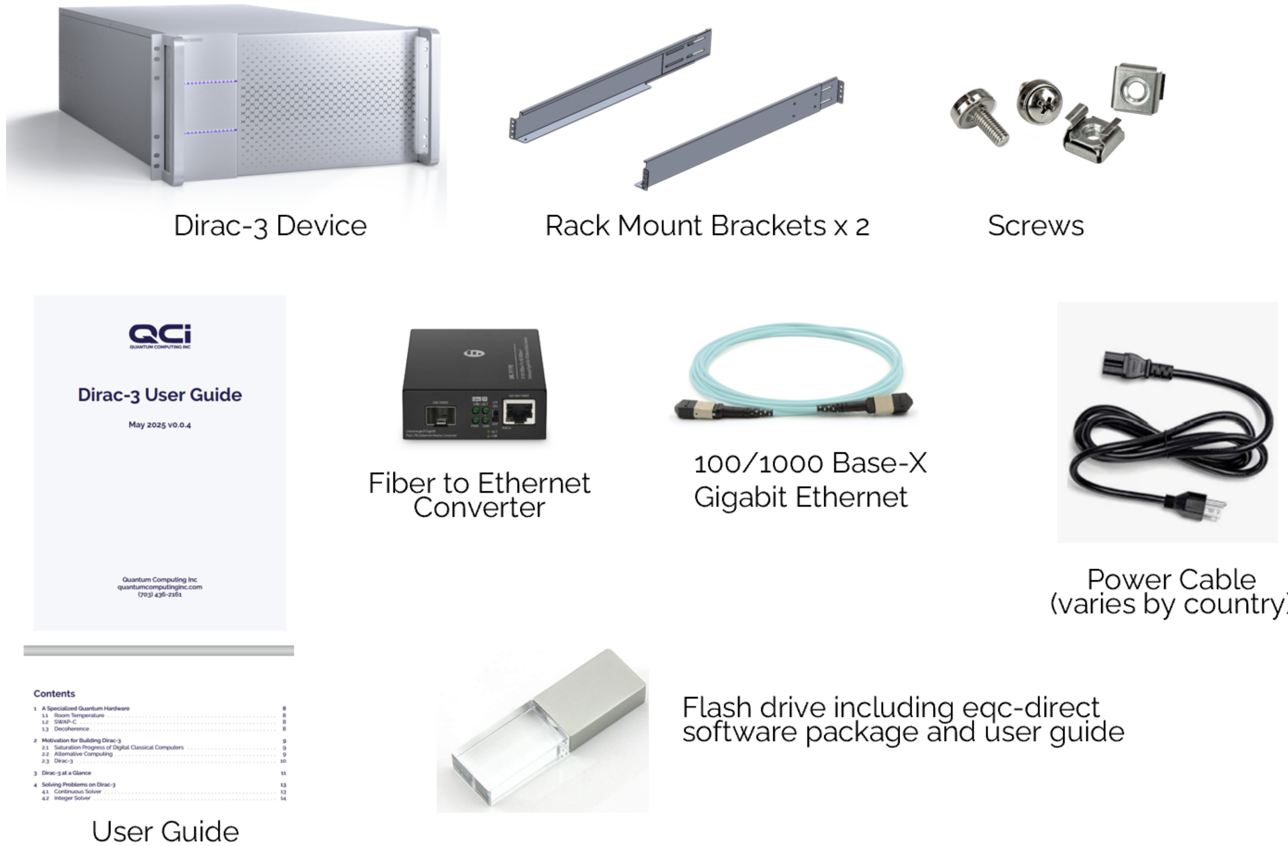


Figure 5: Dirac-3 device and other items included when open package

7 Setting up Dirac-3

7.1 Equipment and Environment Requirements

When setting up your device it is necessary to choose an appropriate location which can meet the following requirements for operation:

- Unrestricted airflow around the server and its vents.
- Temperature must not exceed 27°C around the server.
- Humidity must not exceed 85%.
- All cables must be kept away from sources of electrical noise, such as radios, power lines, and fluorescent lighting fixtures.
- The cable length from a switch to an attached device cannot exceed 328 feet (100 meters) as Ethernet Specified.

7.2 Safety Precautions

⚠The Dirac-3 system is large and heavy! When installing, at least TWO people are required to safely and easily lift and move the unit. Installation with one person is NOT recommended!
⚠Dirac-3 utilizes class 3B/3R Lasers which are dangerous for human eyes! DO NOT open the system!
⚠Do not operate Dirac-3 in an area that exceeds the maximum recommended ambient temperature of 27°C to prevent overheating of the server. Allow at least 3 inches (7.6 cm) of clearance around the ventilation openings to prevent airflow restriction.

Installation of the equipment must comply with local and national electrical codes.

7.3 Server Rack Installation

This section describes the procedure when mounting a Dirac-3 system into a server rack. While not required for normal operation, it is recommended to install the system in a temperature controlled, well ventilated area such as a server room in order to meet operational environment requirements.

Prepare the Device: Ensure the 5U device is powered off and disconnected from power sources. Remove any packaging materials or protective covers from the device.

Attach Server Rack Rails: Attach and secure server rack rails to both sides of the vertical post of the rack. See figure 6 and figure 7

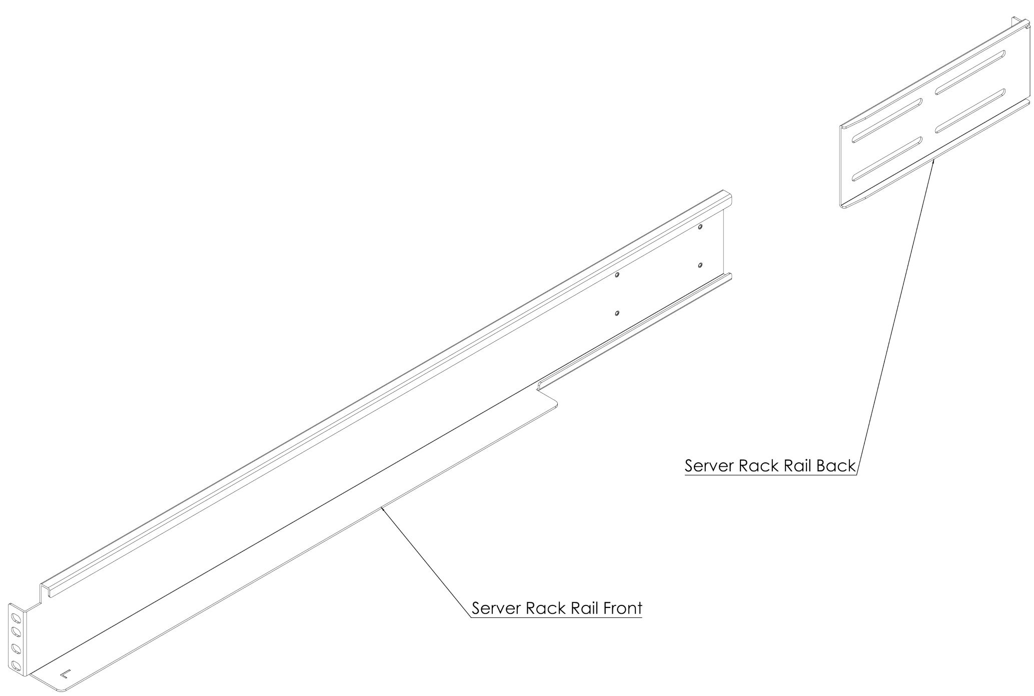


Figure 6: Server rack rails front and back

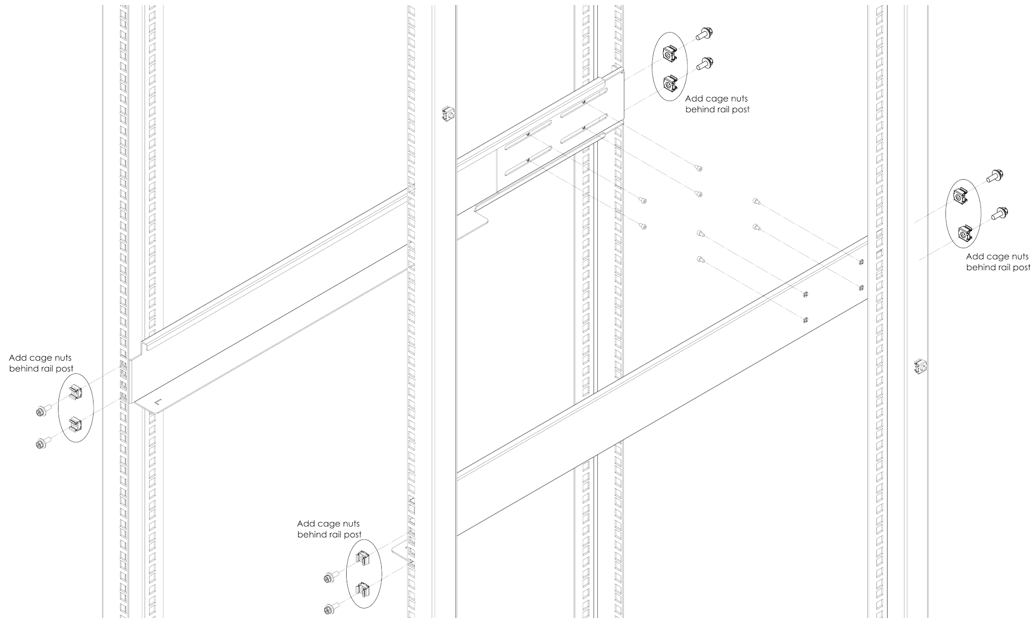


Figure 7: Mounting rails

Position and Secure the Device: Carefully carry the device and gently slide it into the desired 5U slot in the rack. Ensure the position of Dirac-3 device parallel to the floor while moving. Once the device is positioned correctly, use screws to secure the mounting rails to the rail support. See figure 8 and 9. Make sure the device is level and aligned properly within the rack.

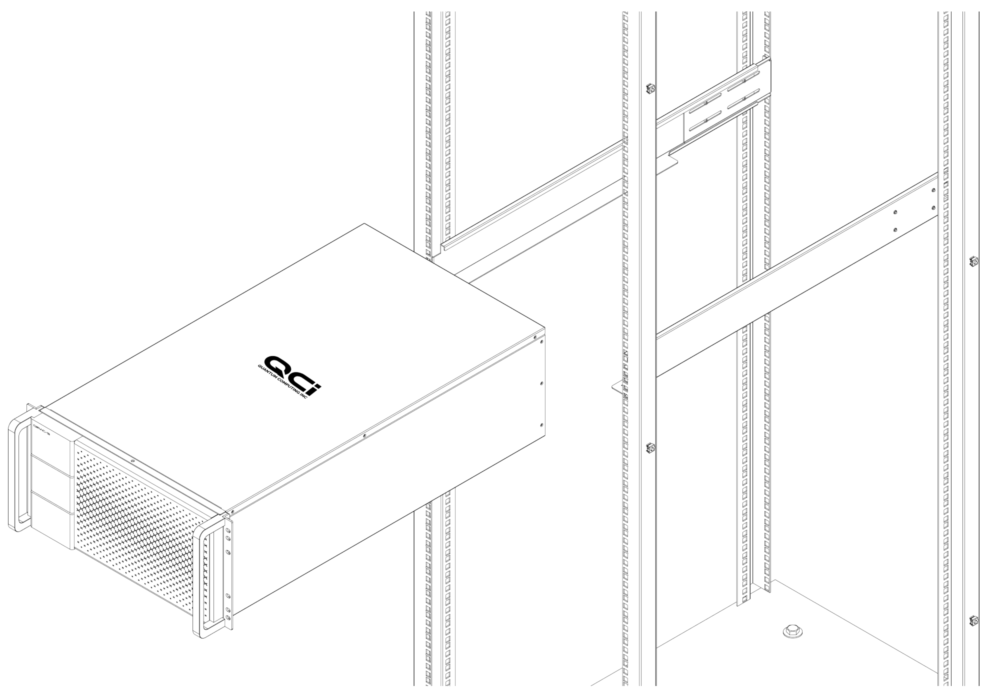


Figure 8: Positioning the Dirac 3

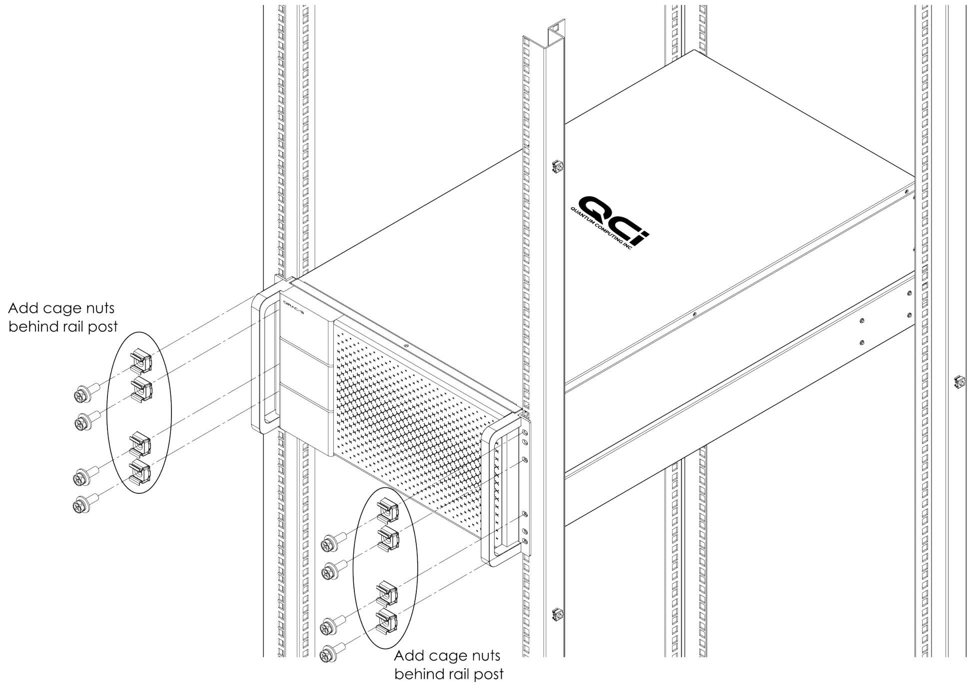


Figure 9: Securing the Dirac 3

Cable Management: Organize power cable and Ethernet cable connections coming from the 5U device to the host server in the rack or outside the rack. Use cable management tools such as cable ties or cable management panels to keep cables neat and organized.

7.4 Powering on Dirac-3

Dirac-3 has been tested and evaluated to operate via direct Ethernet interface with a host computer. For simplicity of integration, this host computer may also be rack mounted with the Dirac-3 system. The host device is recommended to be compatible with Linux operation system. A diagram of the intended setup is provided in figure 11:

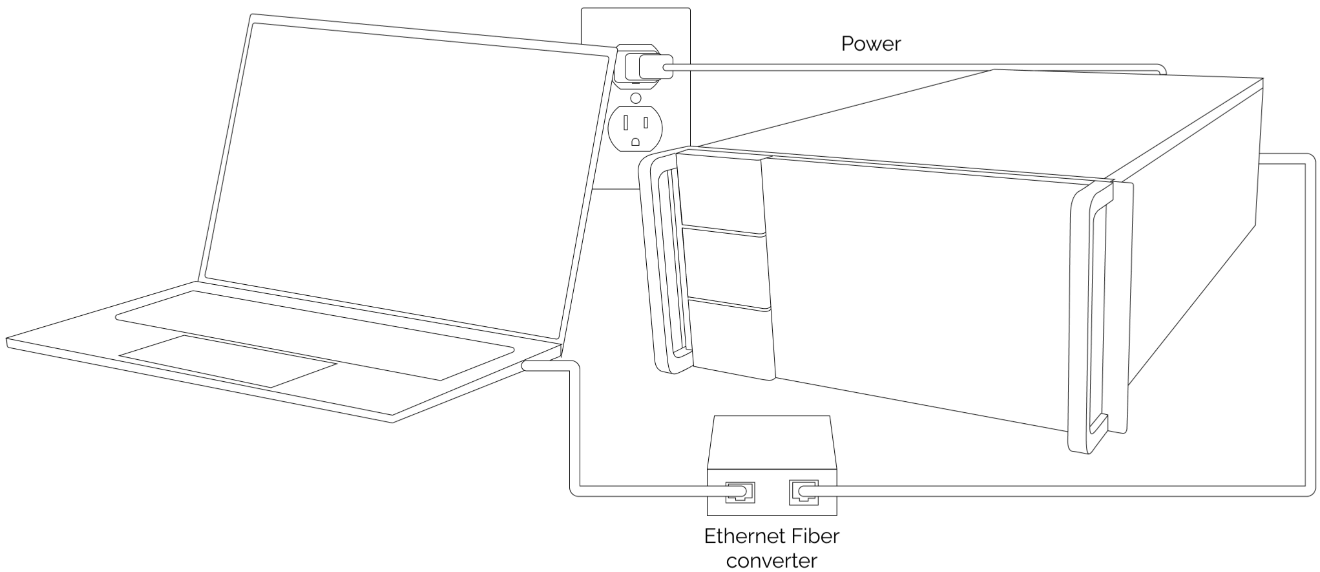


Figure 10: Example of Dirac-3 setup where host computer is connected directly to the device over physical Fiber/Ethernet connection

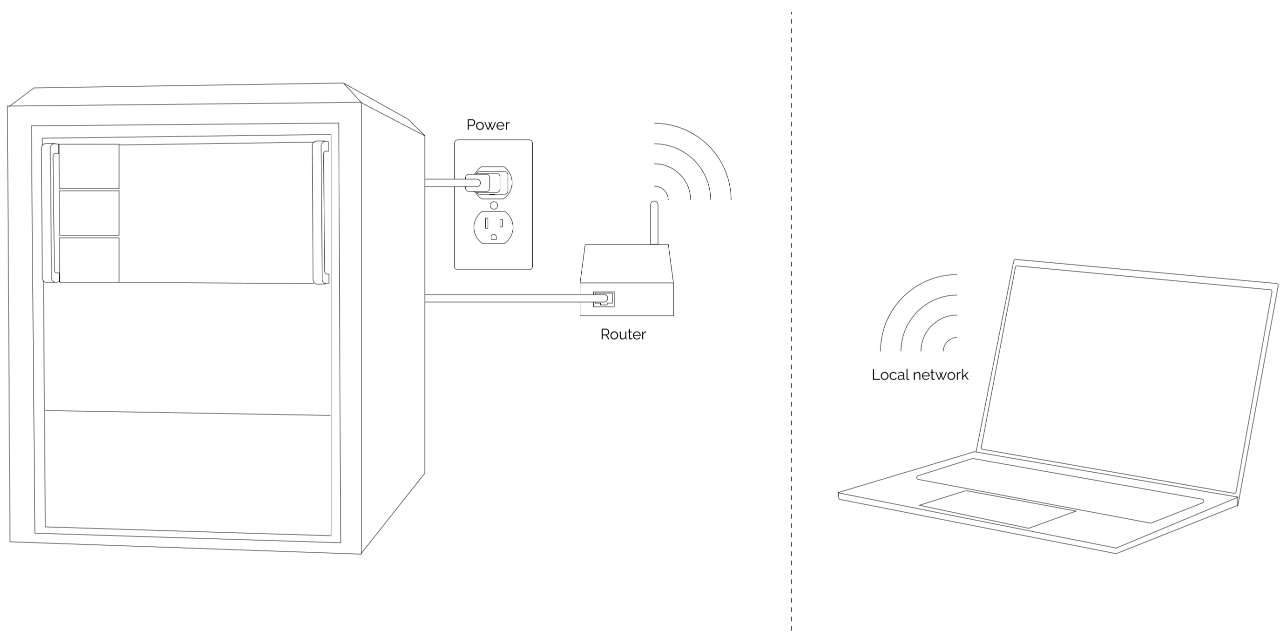


Figure 11: Example of Dirac-3 setup where the device is positioned in a standard data center rack, connected to a router. Users can access the device within local network.

Figure 12 illustrates the input/output connections for Dirac-3. The back panel of Dirac-3 includes an AC power plug port, a switch, an Ethernet port for connecting to the host computer, and a USB port, which is closed and intended for only manufacturer debugging purposes.

⚠Do not use the USB port.

Two LEDs located on the front panel (left) indicate device status.

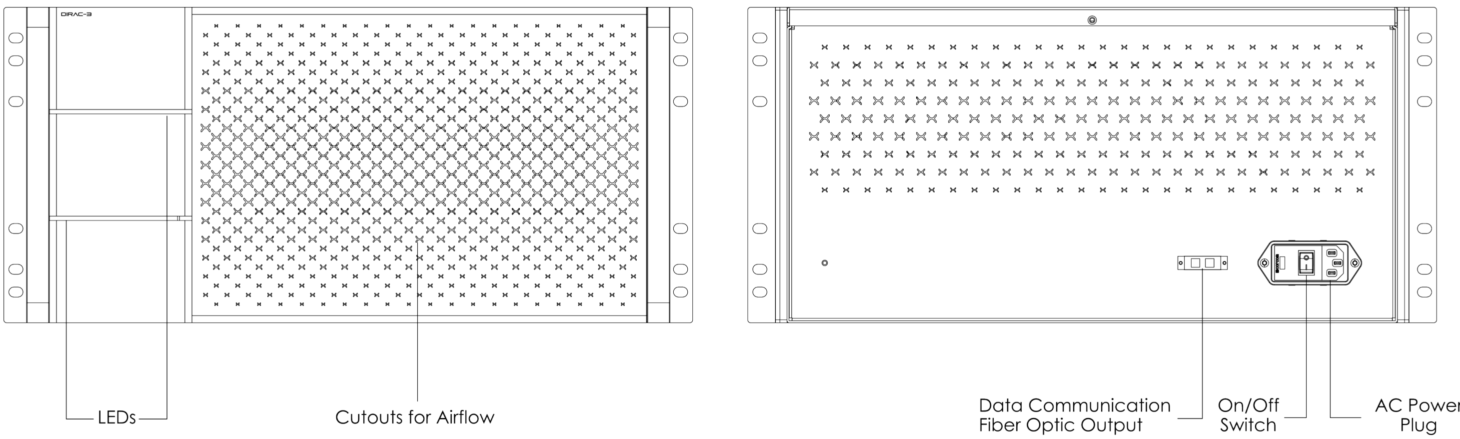


Figure 12: Front (left) and Back (right) panels of Dirac-3.

7.5 Connecting Dirac-3 to a Host Computer

Before plugging the device into a wall outlet or a host device, check to make sure that the power switch is set to "OFF". Once verified, plug the power cord into the unit. Next, plug the power cord into the wall outlet. Power on the device by flipping the power switch.

The Dirac-3 device will be delivered with a static IP address of 192.168.42.13/24. To connect to it, configure your laptop's Ethernet adapter with an IP address in the 192.168.42.x range, where x is an unused address on the same subnet (e.g. not 13, 1, or 0) and the mask to 255.255.255.0. Once these settings have been changed, you may plug the Ethernet cable into the jack, and then into the host computer. In order to verify that the Dirac server is accessible, users must utilize the system monitoring capabilities of the Python package eqc-direct. In order to install *eqc-direct* the latest available version from PyPi run:

```
pip install eqc-direct
```

The package supports Python version 3.8, 3.9, and 3.10. Next, using the Python package follow the instructions in the documentation under the System Monitoring heading to check if your device is up and running. When the device is powered on it begins by entering a calibration process to determine if the system is functioning properly. Using the system monitoring functions, users can check to see when their device is available to begin solving problems.

8 Writing an Initial Program for Dirac-3

Here we share a basic problem submission example for Dirac-3. All interactions with the device are resolved using eqc-direct a Python package which utilizes gRPC messaging to interface with the device. The package provides utility functions such as acquiring a lock on the device, system monitoring, testing the health of the device, submitting a problem, and cancelling a currently processing problem submission. In order to illustrate a standard problem submission a simple polynomial problem will be utilized:

$$f(X) = 3x_4 + 2.1x_1^2 + 1.5x_2^2 + 7.9x_2x_3 + x_2x_4^2 + x_3^3 \tag{9}$$

The first step is to extract the polynomial coefficients and format polynomial variable indices for each term in the equation. The polynomial coefficients will be represented as a list as follows:

```
poly_coefs = [3, 2.1, 1.5, 7.9, 1, 1]
```

The polynomial indices for the coefficients will be represented as follows in the same order as represented in the original equation:

```
poly_indices = [[0,0,4], [0,1,1], [0,2,2], [0,2,3], [2,4,4], [3,3,3]]
```

Note, that each individual polynomial index set has non-decreasing values moving from left to right such that for each set of indices:

$$[k_1, k_2, k_3, \ldots, k_{n-1}, k_n] \tag{10}$$

the following condition must hold:

$$k_1 \leq k_2 \leq k_3 \leq \cdots \leq k_{n-1} \leq k_n \tag{11}$$

This property guarantees the uniqueness of the submitted polynomial index set. Also, this sparse representation minimizes the amount of data needed to be transferred to the device to obtain a solution.

After formatting the problem for submission, the client can be used to submit the problem after acquiring an exclusive execution lock on the device. The inputs to the function used to submit problems, *solve_sum_constrained*, are as follows:

- *lock_id* - a UUID string with exclusive lock for device execution
- *poly_coefficients* - the coefficient values for polynomial to be minimized. Numbers, including integers, should be floats with 32-bit (or less) precision, otherwise precision is lost during conversion to 32-bit.
- *poly_indices* - list of lists containing the indices for coefficient values for polynomial to be minimized.

- *num_variables* - optional input to specify number of variables for polynomial. Must be greater than or equal to maximum index value in *poly_indices*.
- *relaxation_schedule* - four different schedules represented in integer parameter. Higher values reduce the variation in the analog spin values and therefore, are more probable to lead to improved objective function energy for input problem. Accepts range of values in set {1, 2, 3, 4}.
- *sum_constraint* - a normalization constraint that is applied to the problem space that is used to calculate energy. This parameter will be rounded if exceeds float32 precision (e.g. 7-decimal places). Value must be between 1 and 10000.
- *solution_precision* - the level of precision to apply to the solutions. This parameter will be rounded if exceeds float32 precision (e.g. 7-decimal places). If specified a distillation method is applied to the continuous solutions to map them to the submitted *solution_precision*. Input *solution_precision* must satisfy *solution_precision* greater than or equal to *sum_constraint*/10000 in order to be valid. Also *sum_constraint* must be divisible by *solution_precision*. If *solution_precision* is not specified no distillation will be applied to the solution derived by the device.

For additional information see source documentation for function here.

```python
#!/usr/bin/env python

from eqc_direct.client import EqcClient

# MUST FILL IN THE VALUES IN THIS FROM YOUR NETWORKING SETUP FOR THE DEVICE
eqc_client = EqcClient(ip_address="localhost", port="50051")
poly_coefs = [3, 2.1, 1.5, 7.9, 1, 1]
poly_indices = [[0,0,4], [0,1,1], [0,2,2], [0,2,3], [2,4,4], [3,3,3]]
# use lock to prevent other users from taking exclusive access to the device
lock_id, start_ts, end_ts=eqc_client.wait_for_lock()
try:
    result_dict = eqc_client.solve_sum_constrained(
        lock_id=lock_id,
        poly_indices = poly_indices,
        poly_coefficients = poly_coefs,
        relaxation_schedule = 2,
        solution_precision=1,
        sum_constraint = 100)
finally:
    # release lock when finished using the device
    lock_release_out = eqc_client.release_lock(lock_id=lock_id)
```

Each response as obtained in *result_dict* above will include the following fields:

- *err_code* - 0 if problem was solved without any issues, otherwise an integer greater than 0 representing the error that occurred

- *err_desc* - a short description of the error code for the submission
- *preprocessing_time* - time in seconds to validate data and re-format input data for running on the device.
- *runtime* - the time in seconds which the device required to solve the problem
- *postprocessing_time* - runtime for auxiliary computations that occur besides sampling during each sample routine from Dirac hardware including intermediate energy calculations, objective function adjustments, and distillation of solutions.
- *energy* - list of energies for best solution found (float32 precision) for each sample from Dirac hardware
- *solution* - a list of vectors representing the lowest energy solution (float32 precision) for each sample from Dirac hardware
- *distilled_runtime* - runtime for distilling solution from the original device solution
- *distilled_energy* - list of energies for distilled solution for input polynomial (float32 precision) for each sample from Dirac hardware
- *distilled_solution* - a vector representing the solution after the distillation procedure is applied to the original solution derived from the hardware. (float32 precision)
- *num_variables* - specifies the number of variables in the polynomial in solved polynomial.
- *num_samples* - the number of samples (independent solutions) to generated from the device
- *calibration_time* - calibration time is unrelated to execution of the individual sampling for the optimization. This time is from system level interruptions from calibrations that happen at regular intervals to maintain system performance
- *start_job_ts* - nanosecond timestamp marking start of submission to device
- *end_job_ts* - nanosecond timestamp marking time at which results were acquired from device and returned to the user

An example response for the problem above would appear as follows:

```
{'err_code': 0,
 'err_desc': 'Success',
 'preprocessing_time': 0.001842498779296875,
 'runtime': [0.0014460086822509766],
 'postprocessing_time': [0.0004482269287109375],
 'energy': [61329.3710938],
 'solution': [[9.686039, 40.7778511, 14.1097012, 35.4264069]],
 'distilled_energy': [58927.1015625],
 'distilled_solution': [[11.0, 40.0, 14.0, 35.0]],
 'num_samples': 1,
 'num_variables': 4,
 'calibration_time': 0.0,
 'start_job_ts': 1747633699646543114,
 'end_job_ts': 1747633700665424457}
```


To run a problem using the integer solver see overview here, function spec for input function *solve_integer* here, and description of function output here.

9 Understanding Device Status

9.1 System Status

The system status provides information regarding whether or not the system is functioning as expected and also provides information on what operations the system is currently performing.

System Status Name	System Status Code	Description
`IDLE`	0	device is not in use and is available for job submission
`JOB_RUNNING`	1	device is in computing mode
`CALIBRATION`	2	device is running a calibration routine. This process occurs only when the device is in an idle state and will temporarily make the system unavailable for problem submissions
`HEALTH_CHECK`	3	device is running health tests to assess the current hardware performance as well as detect potential issues with the hardware. This process occurs only when the device is in an idle state and will temporarily make the system unavailable for problem submissions
`HARDWARE_FAILURE`	4, 5	Device has entered a failed state and is no longer available to solve problems until hardware issues have been remediated

Table 2: System status information summary.

9.2 Submission/Results Statuses

Every single submission and results object contains fields *err_code* and *err_desc*. A nonzero error code indicates that the submission or result experienced that specific error while processing. If a nonzero error code is present the submission will not return any data from the Dirac-3 device.

Error	Code	Description
`normal_code`	0	Success
`index_out_of_range`	1	Index in submitted data is out of range for specified number of variables.
`coef_index_mismatch`	2	Polynomial indices do not match required length for specified coefficient length.
`device_busy`	3	Device is currently processing another request.
`lock_mismatch`	4	`lock_id` does not match the current device lock.
`hardware_failure`	5	Device failed during execution.
`invalid_sum_constraint`	6	Sum constraint must be ≥ 1 and ≤ 10000.
`invalid_relaxation_schedule`	7	Parameter `relaxation_schedule` must be in the set {1, 2, 3, 4}.
`user_interrupt`	8	User sent stop signal before result was returned.
`exceeds_max_size`	9	Exceeds maximum problem size for device.
`decreasing_index`	10	One or more polynomial indices are not in non-decreasing order.
`invalid_precision`	11	Input precision exceeds the device's maximum allowed precision.
`num_samples_positive`	12	Input `num_samples` must be positive.
`precision_constraint_mismatch`	13	Sum constraint must be divisible by solution precision.

Table 3: Problem submission and results error information summary.

Error	Code	Description
`precision_nonnegative`	14	Input solution precision cannot be negative.
`num_variables_positive`	15	Input `num_variables` must be greater than 0.
`degree_positive`	16	Input degree must be greater than 0.
`num_levels_num_vars_mismatch`	17	Length of `num_levels` must match `num_variables`.
`num_levels_gt_one`	18	All elements of input 'num_levels' must be greater than 1
`total_integer_levels`	19	Total number of integer levels from input variables num_levels exceed limit
`invalid_mean_photon_number`	20	Mean photon number if specified must be in range [0.0000667, 0.0066666]
`invalid_quantum_fluctuation_coefficient`	21	Quantum fluctuation coefficient if specified must be in range [1, 100]

Table 4: Problem submission and results error information summary (continued).

10 Remote Command Line Interface for Administrators

To facilitate routine administrative tasks and enhance device security, QCi allows administrators to connect to the device via SSH. The designated user for administrative tasks is qciadmin, which has access to a custom Command Line Interface (CLI). This interface allows users to modify the device's networking configuration, change passwords, power down, or reboot the device. To access the device, first obtain its current IP address. Before it is reassigned the device is located 192.168.42.13/24. To cancel any input to a function in the CLI, simply press CTRL+C in the terminal. This action will interrupt and cancel the current command before it executes.

10.1 Remote login

Using the IP address you've identified for your device you will be able to SSH to the admin portal from the command line remotely using the following command, which will prompt you for a password which is on a sticker on the back panel of the device:

```
ssh qciadmin@YOUR_DEVICE_IP_ADDRESS
```

After logging in, the user will be presented with a list of available commands to execute. A list of all available commands can be found in the table below:

Input #	Command	Action
0	Exit	Immediately ends user session and disconnects user from uQRNG CLI
1	Power off system	Shutdown system remotely however does not turn off power to the photonic portion must press button back panel to power down
2	Reboot system	Restarts system
3	Show current IP address	Prints current IP address to terminal
4	Change password	Allows user to change qciadmin password
5	Configure network interface	Allows user to modify network interface including: IP address, subnet, gateway, and DNS server
6	Show network interface config	Prints content of ethernet network file to terminal
7	Add a public key for SSH auth	Allows user to add a public key for authentication to the device for SSH
8	Delete a public key for SSH auth	Allows user to delete a public key that was available for authentication to the device for SSH
9	Show all public keys for SSH	Prints all public keys that are currently available for authentication

Table 5: List of all available commands through admin terminal.

10.2 Changing password

It is recommended as a first step to change your password from the default value. To do this, input the number 4 in the terminal and press Enter. To reset your password, you will need to confirm your current password and then enter your new password twice for validation:

```
Enter your choice (0-9): 4
Changing password for qciadmin.
Current password:
New password:
Retype new password:
```

10.3 Changing the network configuration

Warning!!! Changing network configurations can lead to connectivity issues and may render your device unreachable if improperly configured. Improper network configuration may also expose your device to unauthorized access by external users.

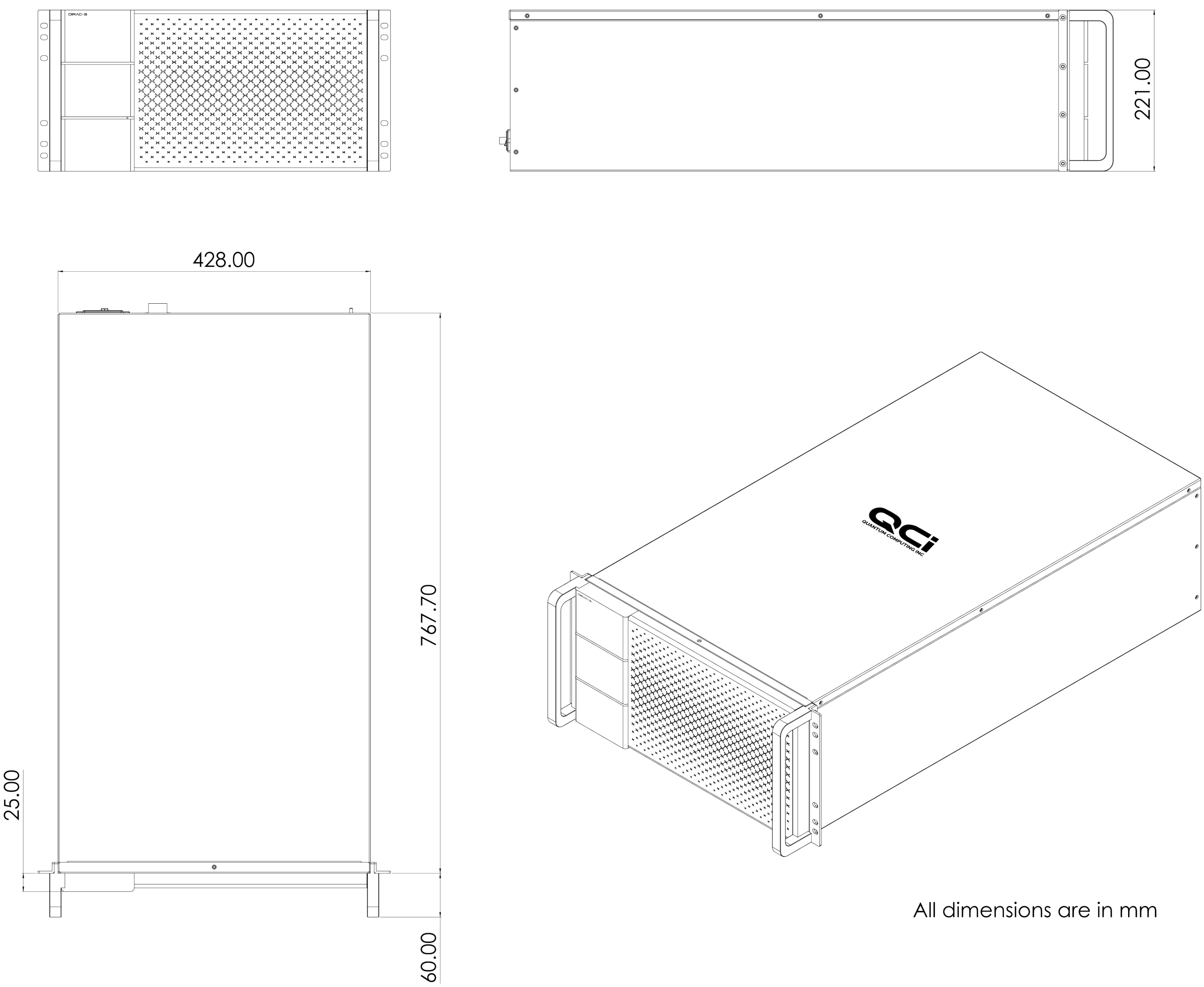
Admin users are allowed to configure the networking of their device using the following attributes:

- **mode** - the operating mode for the Ethernet interface to the uQRNG device can be either static or dhcp
- Additional static configurations:
  - **IP address** - Pv4 formatted IP address
  - **subnet mask** - (CIDR format) 32-bit number used in IP networking to divide an IP address into network and host portions
  - **gateway** - specify device or a node on a network that serves as an access point to another network. It's optional, but if provided, it allows the device to communicate beyond its local subnet. Must be IPv4 formatted.
  - **dns server** - DNS servers within local networks (e.g., corporate networks, home networks) resolve internal domain names that are not publicly accessible on the internet. This capability facilitates efficient communication among devices within the same network. Currently only one can be specified for this input.

An example of using the CLI when specifying static can be seen below:

```
Enter your choice (0-9): 5
Enter the mode (static/dhcp): static
Enter IP address (IPv4): 192.168.42.42
Enter subnet mask (CIDR):
/24
Enter gateway (IPv4, optional, press Enter to skip):
Enter DNS servers (IPv4, optional, press Enter to skip):
```

11 Mechanical



All dimensions are in mm

12 Warranty

For warranty information and support services, please visit our website or contact our customer support team.

13 Troubleshooting and Support

If you encounter any issues with the Dirac-3 device or have questions about its operation, refer to this manual, the software package documentation, or contact our customer support for assistance at support@quantumcomputinginc.com or at https://quantumcomputinginc.ladesk.com/

References

[1] Fabio Furini et al. "QPLIB: A Library of Quadratic Programming Instances". In: *Mathematical Programming Computation* (2018). DOI: 10.1007/s12532-018-0147-4.

[2] Nicolas Gisin et al. "Quantum cryptography". In: *Rev. Mod. Phys.* 74 (1 Mar. 2002), pp. 145–195. DOI: 10.1103/RevModPhys.74.145. URL: https://link.aps.org/doi/10.1103/RevModPhys.74.145.

[3] A.D. King, J. Raymond, and T. et al. Lanting. "Quantum critical dynamics in a 5,000-qubit programmable spin glass". In: *Nature* 617 (2023), pp. 61–66. DOI: https://doi.org/10.1038/s41586-023-05867-2.

[4] Prajnesh Kumar et al. "Efficient reservoir computing using field programmable gate array and electro-optic modulation". In: *OSA Continuum* 4.3 (Mar. 2021), pp. 1086–1098. DOI: 10.1364/OSAC.417996. URL: https://opg.optica.org/osac/abstract.cfm?URI=osac-4-3-1086.

[5] Yi-Xuan Ma and Peng-Bo Li. "Deterministic generation of phononic Fock states via weak nonlinearities". In: *Phys. Rev. A* 108 (5 Nov. 2023), p. 053709. DOI: 10.1103/PhysRevA.108.053709. URL: https://link.aps.org/doi/10.1103/PhysRevA.108.053709.

[6] Lac Nguyen et al. *Entropy Computing: A Paradigm for Optimization in an Open Quantum System*. 2024. arXiv: 2407.04512 [quant-ph]. URL: https://arxiv.org/abs/2407.04512.

[7] R Gary Parker and Ronald L Rardin. *Discrete optimization*. Elsevier, 2014.

[8] M. Uria, P. Solano, and C. Hermann-Avigliano. "Deterministic Generation of Large Fock States". In: *Phys. Rev. Lett.* 125 (9 Aug. 2020), p. 093603. DOI: 10.1103/PhysRevLett.125.093603. URL: https://link.aps.org/doi/10.1103/PhysRevLett.125.093603.

[9] Y. Yamamoto, K. Aihara, and T. et al. Leleu. "Coherent Ising machines—optical neural networks operating at the quantum limit". In: *npj Quantum Inf* (2017). DOI: https://doi.org/10.1038/s41534-017-0048-9.