



Dirac-3 User Guide

March 2024 v0.0.2

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

Contents

1	A Specialized Quantum Hardware	8
1.1	Room Temperature	8
1.2	SWAP-C	8
1.3	Decoherence	8
2	Motivation for Building Dirac-3	9
2.1	Saturation Progress of Digital Classical Computers	9
2.2	Alternative Computing	9
2.3	Dirac-3	10
3	Dirac-3 at a Glance	11
4	Solving Problems on Dirac-3	12
4.1	All-to-All Connectivity	12
4.2	Dynamic Range	12
4.3	Summation Constraint	13
4.4	Solving Over Discrete Spaces	13
4.5	Relaxation Schedules	14
4.6	Analog Nature	15
4.7	Many-body Problems	16
5	Understanding Dirac-3 Performance	17
5.1	Understanding Computing Time	17
5.2	Understanding Power Consumption	19
6	Unpacking Dirac-3	20
6.1	Package Contents	20
7	Setting up Dirac-3	20
7.1	Equipment and Environment Requirements	20
7.2	Safety Precautions	21
7.3	Server Rack Installation	21
7.4	Powering on Dirac-3	22
7.5	Connecting Dirac-3 to a Host Computer	24
8	Writing an Initial Program for Dirac-3	25
9	Understanding Device Status	29
9.1	System Status	29
9.2	Submission/Results Statuses	30
10	Mechanical	31
11	Warranty	31

Overview

Welcome to the Dirac-3 User Guide. Dirac-3 is the third generation of QCI's flagship product line: Entropy Quantum Computing (EQC). EQC is a quantum-hardware approach to solving complex, many-body, and highly interconnected multi-variable optimization problems. This manual provides easy-to-follow instructions for the setup/installation and provides an overview of hardware features.

For guidelines on how to formulate problems that are ideally suited to run on Dirac-3 and to further explore applications, please see the "Dirac-3 Programming Guide".

For a better understanding of the fundamental physics behind QCI's EQC technology and operating principles of Dirac-3, please see "An Open Quantum System for Discrete Optimization".

Notice and Disclaimer

This User Guide, including all text, images, diagrams, and other content contained herein, is the intellectual property of Quantum Computing Inc. All rights, including copyright, are reserved. No part of this User Guide may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language without the prior written permission of Quantum Computing Inc.

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 submit a ticket to <https://quantumcomputinginc.ladesk.com/>.

Warning

⚠ Non-Compliance Notice: This device has not yet passed CE or FCC emissions certification testing, so it might not be compliant with industry standards and regulations. Users should take appropriate precautions and use the device at their own risk.

Future Compliance Updates: QCi is committed to continuous improvement and may 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	<ul style="list-style-type: none">• Section 8 - Updated programming example.• Section 9.2 - Added additional validation failure codes• Section 4.6 - Correct x and y axis labels on Figure 1

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 and systems for solving a large variety of real-world problems. One methodology is called “Entropy Quantum Computing” (EQC).

1.1 Room Temperature

Many quantum technologies which use massive particles (such as electrons, ions, atoms, 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 electromagnetic 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 Saturation Progress of Digital Classical Computers

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 [3].

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 [4], [2], [1]. 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 ($k=1, 2, 3, 4, \dots$) 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

ENTROPY QUANTUM COMPUTING	
Solver Type	Constrained Discrete Number Optimization
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

Order of Correlation	Maximum Number of Variables
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 of objective function minimization and maximization for optimization over discrete spaces by finding the ground state of a complex system with many inter-correlated variables. They correspond to minimizing or maximizing the expected return of the following objective function:

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

under the constraint of a fixed resource R :

$$R = \sum_{i=1}^N V_i \quad (2)$$

where V_i is the value of each variable, C_i is the linear return of each variable which must be real numbers, J_{ij} , T_{ijk} , Q_{ijkl} , P_{ijklm} are joint returns of variables which must be real numbers, and R is the constraint summation satisfying the following inequality:

$$1 \leq R \leq 10000 \quad (3)$$

Dirac-3 allows direct submission for minimization only. It is assumed that users perform a simple transformation before submitting the problem to handle maximization (multiply by -1).

4.1 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.

4.2 Dynamic Range

Unlike an Ising Solver where variables in an integer optimization problem must first be translated into binary ones, Dirac-3 an encoding advantage by utilizing dynamic range allowing direct encoding of integer variables.

Dirac-3 supports a dynamic range from 23 dB up to 40 dB, allowing the ratio between the highest and lowest values of coefficients among C_i , J_{ij} , T_{ijk} , Q_{ijkl} , and P_{ijklm} to range from 200 to 10000. The analog nature of Dirac-3 causes the dynamic range of the system to vary with the temperature fluctuation and individual imperfections of devices when measuring quantum states. Dirac-3 still takes in any dynamic range that is readable by classical digital system (up to 70dB). However, it will struggle to recognize those coupling terms whose values are 23dB or more below the maximum

values of all coupling terms.

It is recommended to formulate the optimization problem where coefficients are 0.005 and above for the best outcome.

4.3 Summation Constraint

The summation constraint described in Equation (2) is a device-related parameter. The maximum possible discrete levels that the device can resolve is 10000. See Table 3 below for more information on how the device's discrete levels and sum constraint are related.

	Minimum	Recommended	Maximum
Sum constraint	1	200 and below	10000

Table 2: Sum constraint recommendation

	Minimum	Maximum
Number of Levels	200	10000
Corresponding Precision Values	$\frac{R}{200}$	$\frac{R}{10000}$

Table 3: Number of discrete levels for each variable V_i or “qudit”

4.4 Solving Over Discrete Spaces

4.4.1 Qudit and Jitter in High-dimensional Schemes

Dirac-3 employs the photonic temporal degree of freedom, specifically utilizing time-bin states of single photons due to their resilience to environmental disturbances, high data transmission rates, and applicability in high-dimensional schemes. The timing jitter of measurement devices defines the bin-width for Dirac-3 qudit.

Dirac-3 aims to solve discrete number optimization instead of conventional binary optimization, as the search space of these two types are drastically different as the number of variables increases.

4.4.2 Shot Noise

After measuring the quantum states of the photons in multiple feedback loops, values of V_i are determined by looking at the detection events of single photons in each time bin. Thus, shot noise originating from the quantum state of light comes from the Poisson statistic that appear in the measurement results.

4.4.3 Continuous vs. Integer

The discrete characteristics of Dirac-3 output stems from single-photon counting and the timing jitter of measurements along the temporal degree of freedom. Given that each V_i has a wide range of possible levels that it could collapse into, the solution outcome from Dirac-3 could produce integer or floating point outputs, depending on the user's choice of the sum constraint and precision parameters.

For example, the following function was run on Dirac-3 to find the minimum:

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

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

$$\begin{aligned} \text{Energy} &= 4379.86 & (5) \\ V_i &= [8.53924, 10.9957, 14.7423, 21.9537, 43.7687]; \end{aligned}$$

The same problem with a higher constraint will return a lower precision from the user's perspective because the number of levels stay the same. For example, Hamiltonian (4) with sum constraint 1000 returns:

$$\begin{aligned} \text{Energy} &= 437956 & (6) \\ V_i &= [87.5832, 109.487, 145.987, 218.989, 437.953]; \end{aligned}$$

The discrete nature of Dirac-3 allows multiple possible levels for each variable. Although solutions can appear continuous, they are representations of the discrete levels the device is solving over. To generate solutions of user-specified precision, a classical Monte-Carlo search is implemented inside the Data Processing Unit of Dirac-3. This is called the distillation process. The possible precision configuration allowed is 0.0001, 0.001, 0.01, 0.1, and 1. When precision is 1, the answers are integers. Users can use this feature or write their own search method to find integer solution as desired. For example, running the same Hamiltonian (4) with sum constraint 100 and precision of 1 returns:

$$\begin{aligned} \text{Energy} &= 4381 & (7) \\ V_i &= [9, 11, 14, 22, 44]; \end{aligned}$$

4.5 Relaxation Schedules

Dirac-3 offers four relaxation schedules, each determining the time allocated for the system to evolve and converge towards the ground state. Schedules 1, 2, 3, and 4 correspond to different time settings, with higher schedule numbers indicating slower system evolution and, consequently, a higher probability of obtaining favorable results. Figure 2 shows the energy distribution of a non-convex problem. After 50 runs, schedule 3 returns optimal answer with higher probability than schedule 2 with the same amount of runs.

4.6 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:

$$E = -10x_1 - 10x_3 - 4x_2^2 - 5x_1x_3 \quad (8)$$

had the following energy distribution in Figure 1 when running on Dirac-3 with a sum constraint of 6 after 100 trials.

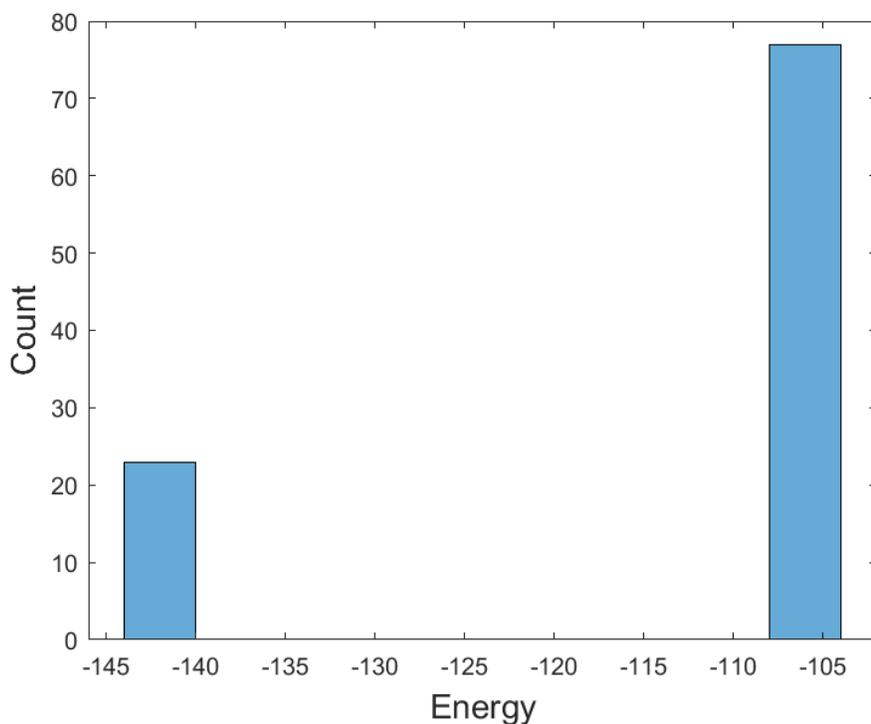


Figure 1: Example of a typical energy distribution results of objective function 8. Dirac-3 returns optimal solution 23% and sub-optimal 77% after 100 trials.

In Figure 2, 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.

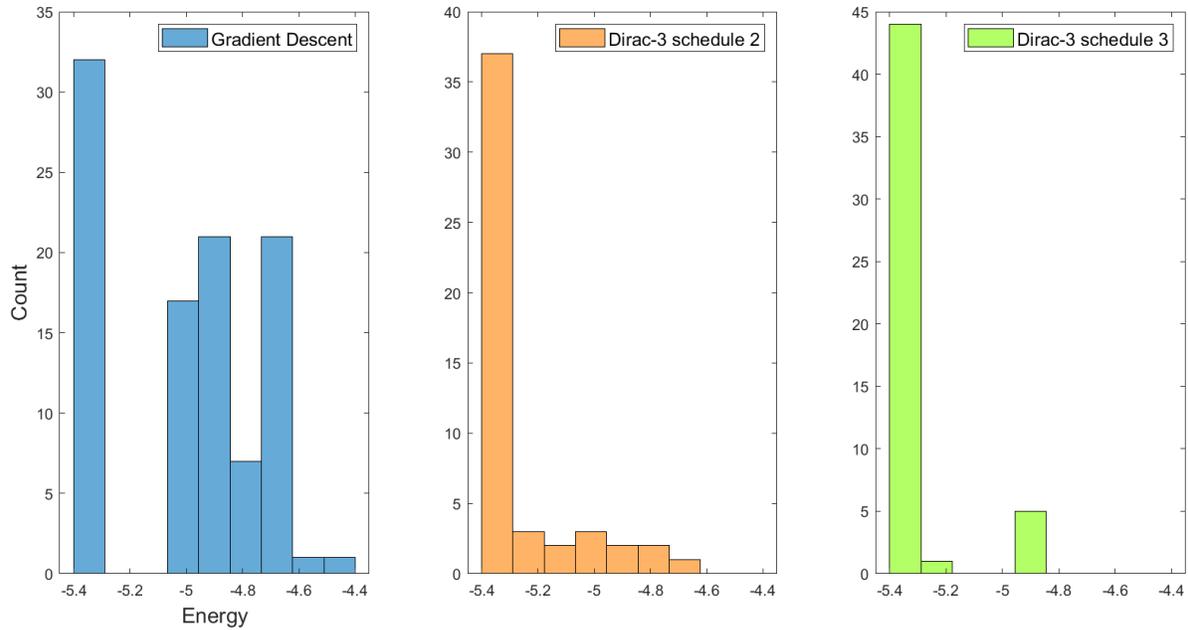


Figure 2: A typical distribution of energy over 50 runs. Dirac-3 performs superior to Classical Gradient Descent, returning optimal answer with high probability.

4.7 Many-body Problems

Dirac-3 goes beyond second-order correlations among variables, striving to tackle challenges found in domains like quantum chemistry, space research, logistics, and beyond. Many-body optimization problems demand substantial computational resources. Figure 3 shows the number of interaction terms among variables versus problem size.

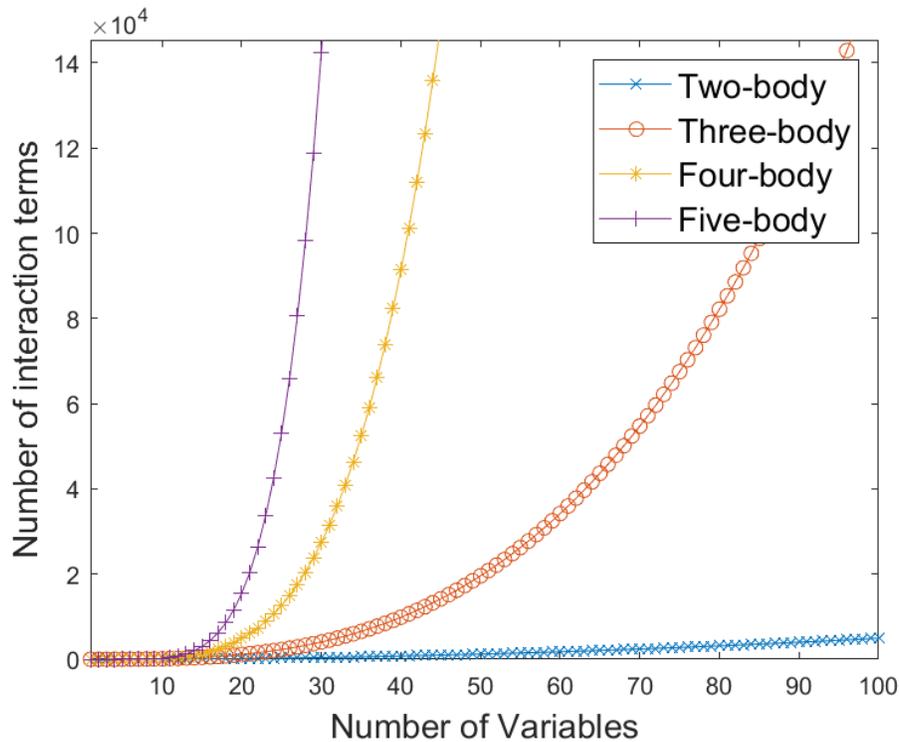


Figure 3: Number of Interaction Terms vs. problem size in a many-body optimization problem

5 Understanding Dirac-3 Performance

5.1 Understanding 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.

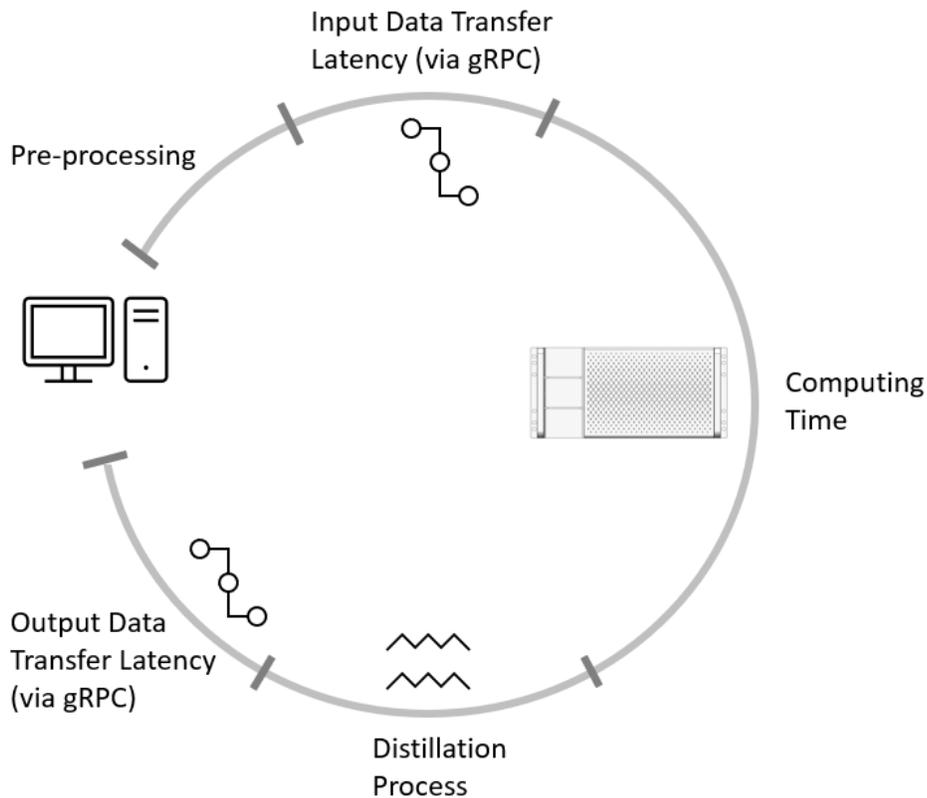


Figure 4: Computing time breakdown in Dirac-3 when it is connected to a single host computer

For users who prefer an integer solution over the raw output provided by Dirac-3, a simple classical distillation process is initiated. This process converts the original continuous solution into an integer solution. Additionally, prior to running a problem the submitted problem is validated and is transformed from the submission format to the matrix representation that is used to solve the problem on the device. The total computing time in this scenario encompasses both the quantum processing time, involving multiple iterations, and the classical search time required for the distillation process, as well as the data validation and polynomial submission format to matrix transformation.

Figure 4 illustrates the breakdown of the total time users experience when running a problem with a host server/computer connected to Dirac-3. Initially, a pre-processing step validates that the data does not exceed the maximum requirements of Dirac-3, followed by mapping the user's input into a format suitable for Dirac-3, specifically symmetric coupling terms. Next, this matrix information is transmitted to the device and computation occurs. After computation, the optional distillation process takes place, followed by transferring the results back to the host computer via the same gRPC interface.

5.2 Understanding Power Consumption

Testing was performed with Dirac-3 to characterize system-level power consumption while solving a problem.

System power was routed through a Chroma 66205 digital power meter, which contains both a voltage meter that sits in parallel with the system, as well as a current meter. The equipment setup is detailed in figure 5. These two measurements combined provide real time power measurements for analysis. Based on these measurements, the power profile for a given problem set is broken down into four main modes of operation:

1. Calibration
2. Initialization (pre-processing and loading of problem into the device)
3. Execution (device perform computing)
4. Return (solution distillation and output)

Power consumption of Dirac-3 is primarily from supporting electronics (fans, amplifiers, drivers, temperature controllers, etc.) and lasers. While executing, Dirac-3 draws additional power due to increased switching currents from digital matrix multiplication in the FPGA and as well as increased signal output from single photon detectors. The rest of the passive optical components and non-linear crystals do not need electrical powering, making Dirac-3 inherently power efficient. An example power profile is shown for a 900-variable problem in table 4.

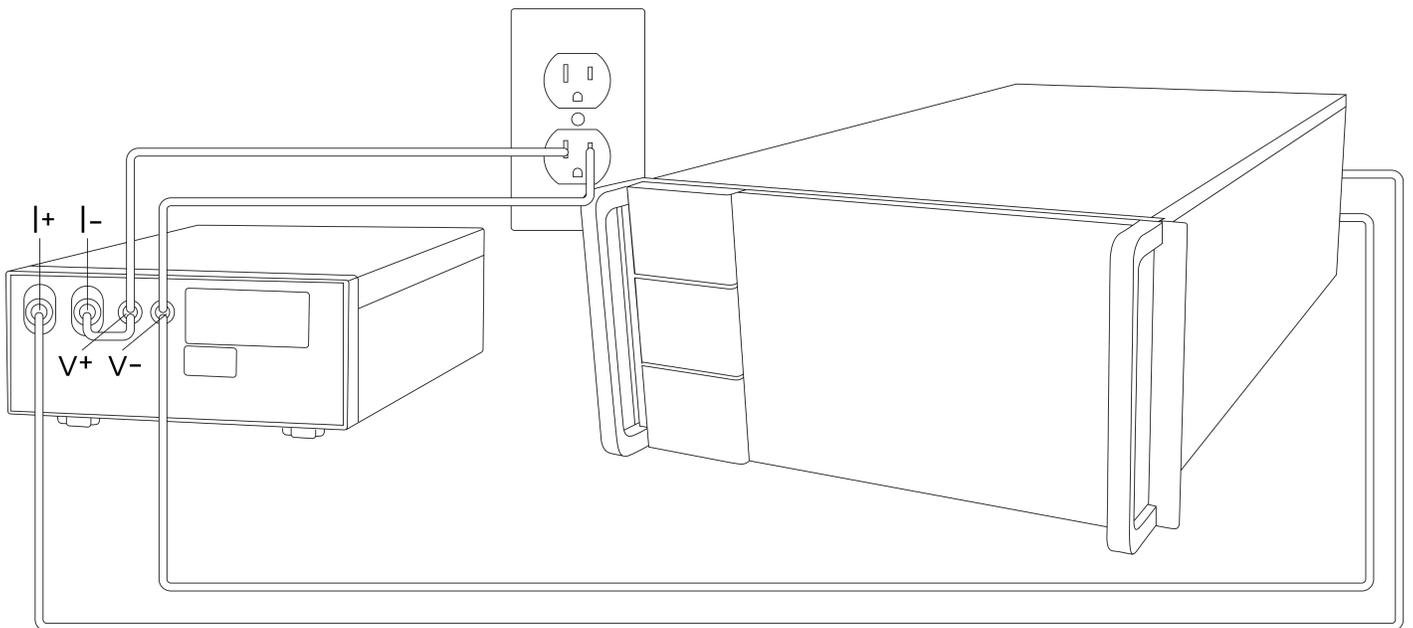


Figure 5: Example of the test setup for measuring Dirac-3 power consumption perform in laboratory

Mode of Operation	Power Draw (W)	Additional Power Draw (W)
Idle	23.31216	0
Calibration	24.79314	1.48098
Initialization	24.04742	0.73526
Execution	23.39218	0.08002
Return	23.56690	0.25474

Table 4: Measured power consumption levels for each mode of operation and compared the additional power draw to the Idle mode

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 content" section. If any item is missing or damaged, contact your QCi Support representative or reseller for instructions.

6.1 Package Contents

- Dirac-3 Device
- Rack Mount Brackets x 2
- Screws
- Ethernet Shielded Cable
- AC Power Cord (will vary by country)
- Documentation: Test Report, User Guide, Programming Guide
- Flash Drive: eqc-direct software package, User Guide, Programming Guide

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

⚠ 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 Rack Mounting Rails: Attach and secure mounting rails to both sides of the device. See figure 6

Attach Rails Support: Attach and secure rail supports to the rack's vertical mounting rails.

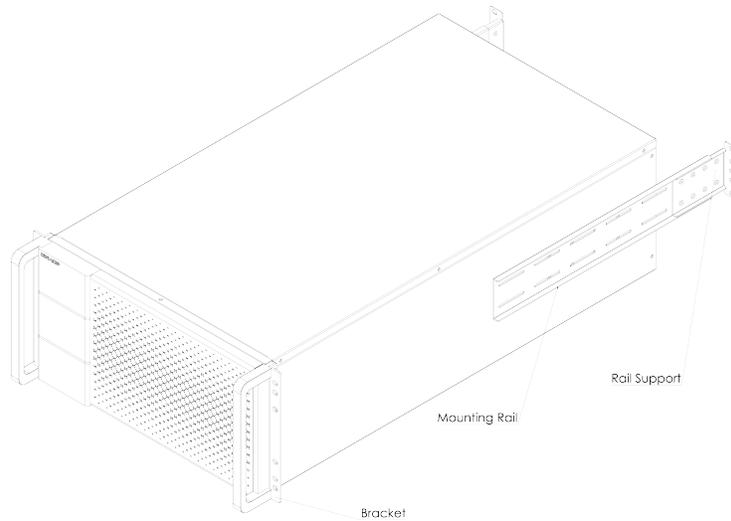


Figure 6: Mounting rails on Dirac-3

Position the Device: Carefully carry the device and gently slide it into the desired 5U slot in the rack. Ensure to position Dirac-3 device parallel to the floor while moving. Align the mounting rails with the corresponding rail support slots or holes attached on the rack's vertical mounting rails.

Secure the Device: Once the device is positioned correctly, use screws to secure the mounting rails to the rail support. Use screws to secure front brackets to the rack. Make sure the device is level and properly aligned within the rack.

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 7:

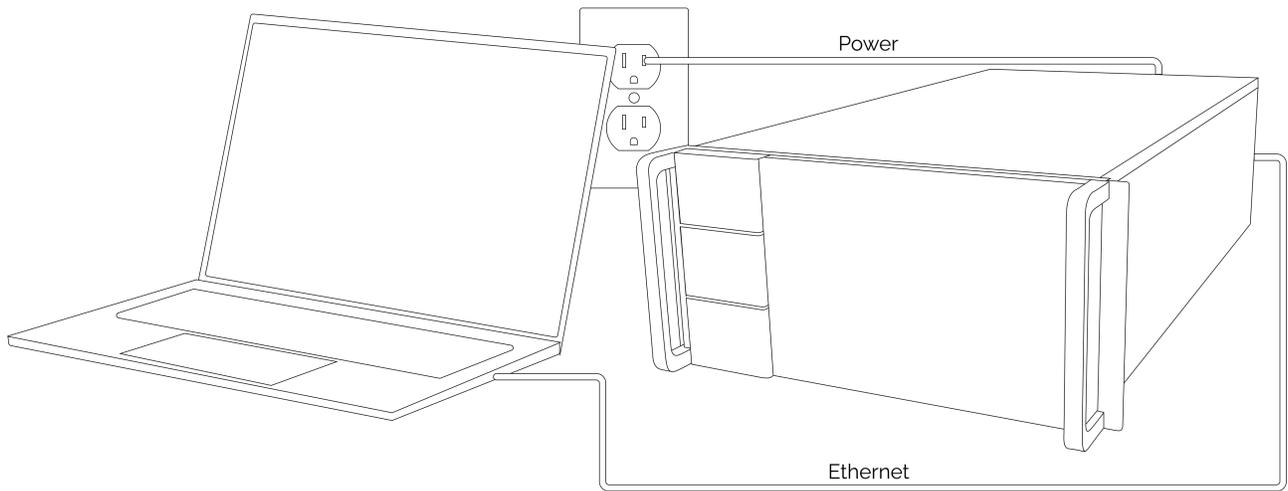


Figure 7: Sketch of Dirac-3 connected via Ethernet interface to a desktop PC, and a power cord going to a wall jack

Figure 8 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. See table 5 for details.

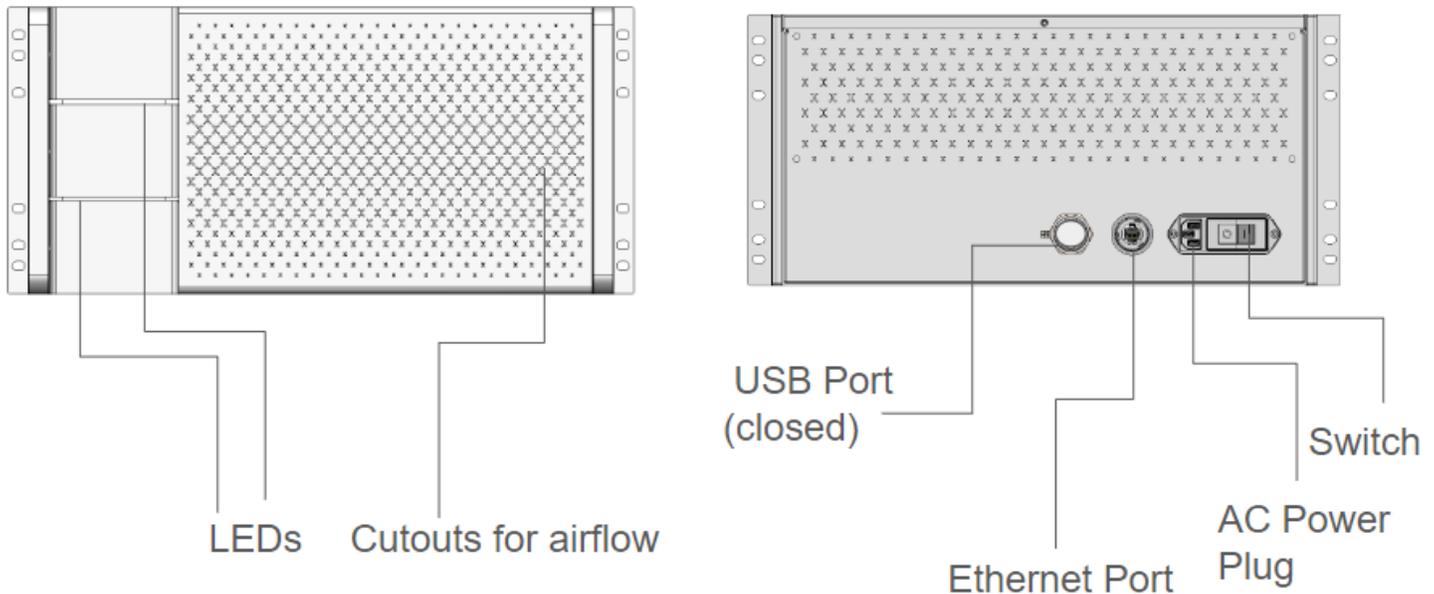


Figure 8: Front (left) and Back (right) panels of Dirac-3.

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. Notice that the LED bar in the front of the unit should be glowing red to indicate power is on and the device is waiting for communication.

Before connecting the device to your computer, set the Ethernet adapter you plan to use to a static IP address (will be provided with device), and set the netmask 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. When ethernet connection has been successfully verified with the host device, the LED indicators will turn green. 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* on your host computer simply pip install the wheel file provided in the flash drive with the following command:

```
pip install eqc_direct_{version_number_}-py3-none-any.whl
```

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.

When the device is busy running computation, LEDs indicator becomes purple. If the LED indicator becomes red, that means that Dirac-3 experience hardware failure or communication failure. Each gRPC request has a status and description in the response that can be used to verify any issues with the device. For additional information see sections [Submission/Results Statuses](#) and [System Status](#).

LED color	Indication
Red	Device is on, no communication with the host computer
Green	Communication connection established between the device and the host computer
Purple	Device is busy (either in computing mode or calibration mode)

Table 5: LEDs indication vs. Device status

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:

$$E = 3x_4 + 2.1x_1^2 + 1.5x_2^2 + 7.9x_2x_3 + x_2x_4^2 + x_3^3 \quad (9)$$

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:

$$[k_1, k_2, k_3, \dots, k_{n-1}, k_n] \quad (10)$$

the following condition must hold:

$$k_1 \leq k_2 \leq k_3 \leq \dots \leq k_{n-1} \leq k_n \quad (11)$$

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, *process_job*, are as follows:

- *lock_id* - a UUID string with exclusive lock for device execution
- *poly_coefficients* - list of coefficient values for the polynomial to be minimized
- *poly_indices* - list of lists containing polynomial indices associated with coefficient values for problem to be optimized
- *num_variables* - optional parameter that supplies the number of total variables for the submitted polynomial must not be less than max index in *poly_indices*. If no value is provided then will be set to max value in *poly_indices*.
- *relaxation_schedule* - four different schedules represented by an integer parameter. Higher values reduce the variation in the analog spin values and therefore lead to a better ground state for input problem. Accepts one of the values in the set {1, 2, 3, 4}.
- *sum_constraint* - a normalization constraint that is applied to the problem space meaning that solution variables from device must sum to provided value. Value must be between 1 and 10000.
- *solution_precision* - optional parameter that specifies the level of precision to apply to the solutions. If specified a distillation method is applied to the continuous solutions to map them to the submitted *solution_precision*. *R* must be divisible by *solution_precision*. Also, *solution_precision* must meet following condition:

$$\frac{R}{\text{solution_precision}} \leq 10000 \quad (12)$$

See sections [Summation Constraint](#) and [Continuous vs. Integer](#). If *solution_precision* is not specified no distillation will be applied to the solution derived by the device.

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

from eqc_direct.client import EqcClient
# MUST FILL IN THE VALUES HERE USING YOUR NETWORKING SETUP FOR THE DEVICE
eqc_client = EqcClient(ip_address="YOUR DEVICE IP ADDRESS",
                      port="YOUR DEVICE PORT")
# using lock prevents other users from attempting to gain exclusive access to the device
lock_id, start_ts, end_ts=eqc_client.wait_for_lock()
result_dict = eqc_client.process_job(
    lock_id=lock_id,
    poly_indices = poly_indices,
    poly_coefficients = poly_coefs,
    relaxation_schedule = 2,
    solution_precision = 1,
    sum_constraint = 100)
# release lock when finished
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
- *energy* - objective value for best *solution* returned by the device
- *solution* - a vector representing the solution to the problem from a given run on the Dirac hardware
- *distilled_runtime* - runtime for distilling solution from the original device solution
- *distilled_energy* - the energy for the distilled solution that was derived from the original solution found by the Dirac hardware
- *distilled_solution* - a vector representing the solution that was derived from the hardware via the distillation procedure.
- *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': 0.0000453,  
'runtime': 1.2345,  
'energy': 12924.37675,  
'solution': [73.25, 16.75, 3.45, 6.55],  
'distilled_runtime': 1.3456,  
'distilled_energy': 12908.3,  
'distilled_solution': [73, 17, 3, 7],  
'start_job_ts': 1710303341382824075,  
'end_job_ts': 1710303342589755079}
```

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, 6, 7	Device has entered a failed state and is no longer available to solve problems until hardware issues have been remediated

Table 6: 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 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 currently processing another request
lock_mismatch	4	lock_id doesn't match current device lock
hardware_failure	5	Device failed during execution
invalid_sum_constraint	6	Sum constraint must be greater than or equal to 1 and less than or equal to 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 max problem size for device
decreasing_index	10	One of specified polynomial indices is not specified in non-decreasing order
invalid_precision	11	the input precision is out of range based on the allowed precision
duplicate_index	12	A duplicate polynomial index set was specified for the input polynomial
precision_constraint_mismatch	13	Sum constraint must be divisible by solution precision
precision_nonnegative	14	Input solution precision cannot be negative
degree_positive	15	Input degree must be greater than 0
num_variables_positive	16	Input num_variables must be greater than 0

Table 7: Problem submission and results error information summary.

12 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 <https://quantumcomputinginc.ladesk.com/>

References

- [1] 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>.
- [2] 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>.
- [3] R Gary Parker and Ronald L Rardin. *Discrete optimization*. Elsevier, 2014.
- [4] 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>.