Q Assets Compositor™: an easy path of compose quantum annealing solutions

Authors

Ezequiel Murina, aQuantum, ezequiel.murina@alhambrait.com
Jose Luis Hevia, aQuantum, jluis.hevia@alhambrait.com
Guido Peterssen, aQuantum, guido.peterssen@alhambrait.com
Mario Piattini, aQuantum/UCLM, mario.piattini@uclm.es

Topic(s)

Quantum software, quantum annealing, quantum software development

Background

Among the different quantum computing approaches, quantum annealing is one of the most promising in the short term, since the first proposal of quantum annealing from Kadowaki and Nishimori in 1998 [1]. In fact, Quantum technology is currently preferentially used to solve combinatorial optimization problems such as, for example, portfolio management, scheduling challenges, optimization, etc. These problems are usually formulated as QUBO (Quadratic Unconstrained Binary Optimization) problems [2], which allows them to be solved very efficiently.

These types of models are used in quantum annealing (based on superconducting circuit, such as D-Wave systems [3]) and digital annealing (based on digital circuits, such as Fujitsu Digital Annealer [4]). It should be noted that, in addition to D-Wave and Fujitsu, the annealing approach increasingly highlights the proposals of IARPA [5], NEC [6] and Qilimanjaro [7].

However, quantum annealing development platforms do not offer user-friendly interfaces for the definition of annealing problems. In fact, much of the existing research in the field of quantum annealing has focused on the needs of the quantum physicist or the quantum mathematician [8] but has not considered the needs of the quantum software developer [9].

In this paper we present a solution to this problem, the Q Assets Compositor™, which is part of QPath® [10], a quantum software development platform to support the design, implementation, and execution of quantum software applications. It is based on a hybrid model for the construction of services that abstract quantum technology without having to worry directly about the manufacturers' platforms and their requirements. QPath® rigorously applies the principles of quantum computing and programming, the good practices in Quantum Software Engineering and Programming [11] and, of course, the

requirements defined by manufacturers for their ecosystems. So that QPath guaranties truly agnostic quantum software development, the compatibility with the ecosystems of different quantum providers, and full and ongoing interoperability with them.

Presentation

The Q Assets Compositor™ facilitates the definition and execution of annealing algorithms in both quantum annealing and digital annealing.

Once all the parameters, variables and rules have been defined through a special intuitive and user-friendly interface, see Figure 1, the systems proceed to compilation and transpilation.

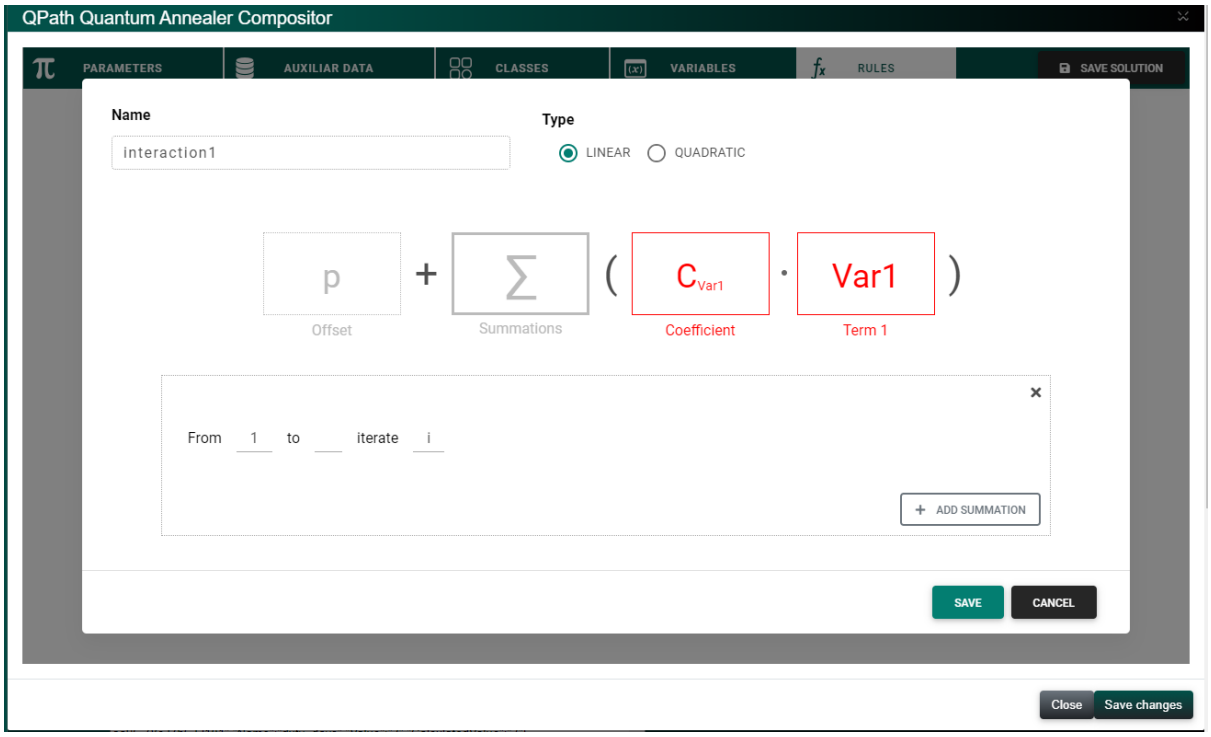


Figure 1.- Interface of QPath´s Q Assets Compositor™ for annealing.

The compilation of the solution, according to a specific syntax, produces a representation that formalize the problem. The transpilation of the metalanguage to Python generates the necessary artifacts, modules, and code to be executed into the final hardware.

QPath® transparently creates all the assets that are needed to the execution, taking care about all the things needed to create complete modules that covers the business layers needs. We chose in the runtime dashboard the specific quantum machine or simulator that acts as an annealing solver.

We are convinced that quantum computing offers us the possibility of initiating a new software engineering golden age [12, 13]; but to achieve this, it is necessary to devise techniques and tools that support the software engineer's task of defining quantum

applications. We think that the QPath´s Q Assets Compositor™ is an important step in this direction.

References

1. Kadowaki, T. and Nishimori, H. Quantum annealing in the transverse Ising model. Phys. Rev. E 58, 5355 –1 November 1998.
2. Glover, F., Kochenberger, G. and Du, Y. A Tutorial on Formulating and Using QUBO Models. https://arxiv.org/abs/1811.11538, 2019. Last revised 13rd July 2021.
3. D-Wave. D-Wave System. https://www.dwavesys.com/ Accessed July 13rd, 2021.
4. Fujitsu. Fujitsu Digital Annealer. https://www.fujitsu.com/global/services/business-services/digital-annealer/index.html Accessed July 13rd, 2021.
5. IARPA. https://www.iarpa.gov/index.php?option=com_content&view=article&id=564&Itemid=339. Accessed July 13rd, 2021.
6. NEC. https://www.nec.com/en/global/quantum-computing/index.html Accessed July13rd 2021.
7. Qilimanjaro. https://www.qilimanjaro.tech/technology/ Accessed July 13rd 2021.
8. Corcoles, A. D., A. Kandala, A. Javadi-Abhari, D. T. McClure, A. W. Cross, K. Temme, P. D. Nation, M. Steffen, J. M. Gambetta (2019). Challenges and Opportunities of Near-Term Quantum Computing Systems. arXiv.org > quant-ph > arXiv:1910.02894.
9. Piattini, M. Requirements for a Robust Quantum Software Development Environment. Cutter Business Technology Journal 34 (4), 12-17, 2021.
10. QuantumPath. https://www.quantumpath.es/ Accessed July 13rd, 2021.
11. Piattini, M., Peterssen, G., Pérez-Castillo, R., Hevia, J.L., Ezequiel Murina, et al. The Talavera Manifesto for Quantum Software Engineering and Programming. QANSWER 2020 QuANtum SoftWare Engineering & pRogramming. Proceedings of the 1st International Workshop on the QuANtum SoftWare Engineering & pRogramming, Talavera de la Reina, Spain, February 11-12, 2020. http://ceur-ws.org/Vol-2561/paper0.pdf
12. Piattini, M., Serrano, M., Pérez-Castillo, R., Peterssen, G. and Hevia, J.L. Towards a Quantum Software Engineering. IT Professional, vol. 23, no. 1, pp. 62-66, 1 Jan.-Feb. 2021, doi: 10.1109/MITP.2020.3019522.

13. Piattini, M., G. Peterssen, and R. Pérez-Castillo. Quantum Computing: A New Software Engineering Golden Age. ACM SIGSOFT Software Engineering Notes, Vol. 45, No. 3, 2020.

Poster

"Q Assets Compositor™: an easy path of composing quantum annealing solutions"

Jose Luis Hevia, Ezequiel Murina, Guido Peterssen, Mario Piattini

Problem

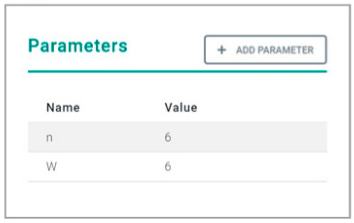
Annealing techniques are used for sampling in Ising model-based optimizations. They require the definition of a square matrix whose dimensions increase with the number of variables of the computational problem. In the code implementation of a high dimensional matrix, we deal with multiple indexes and mathematical expressions that include them. It is a hard task. Nowadays, there is a lack of software tools that make easier the work of the programmers in this scenario.

Solution

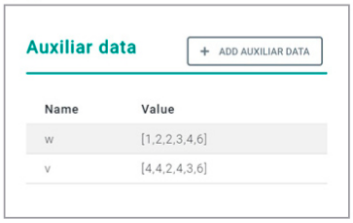
The Q Assets Compositor™ included in QuantumPath® is a user-friendly interface that makes easier the definition of annealing problems. It offers different tools that cover all the stages on an annealing solution: data input, set up of parameters, definition of variables and correlations.

Simplifying annealing problem solutions with the Q Assets Compositor™

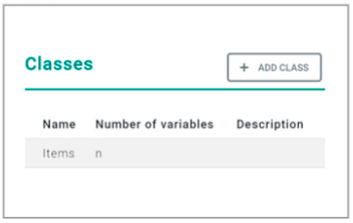
Step 1



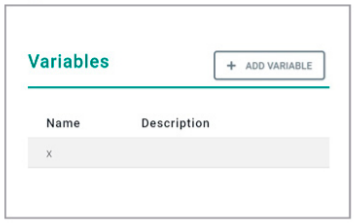
Step 2



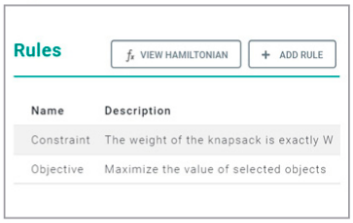
Step 3



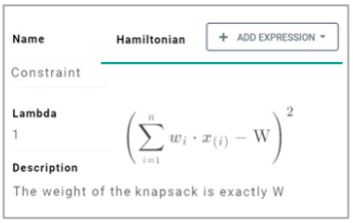
Step 4



Step 5



Step 6



Try Quantum Path®
FREE DEVELOPER SUBSCRIPTION

aquantum.es

quantumpath.es

Scientific/Technical Talk at EQTC 2021, Ireland