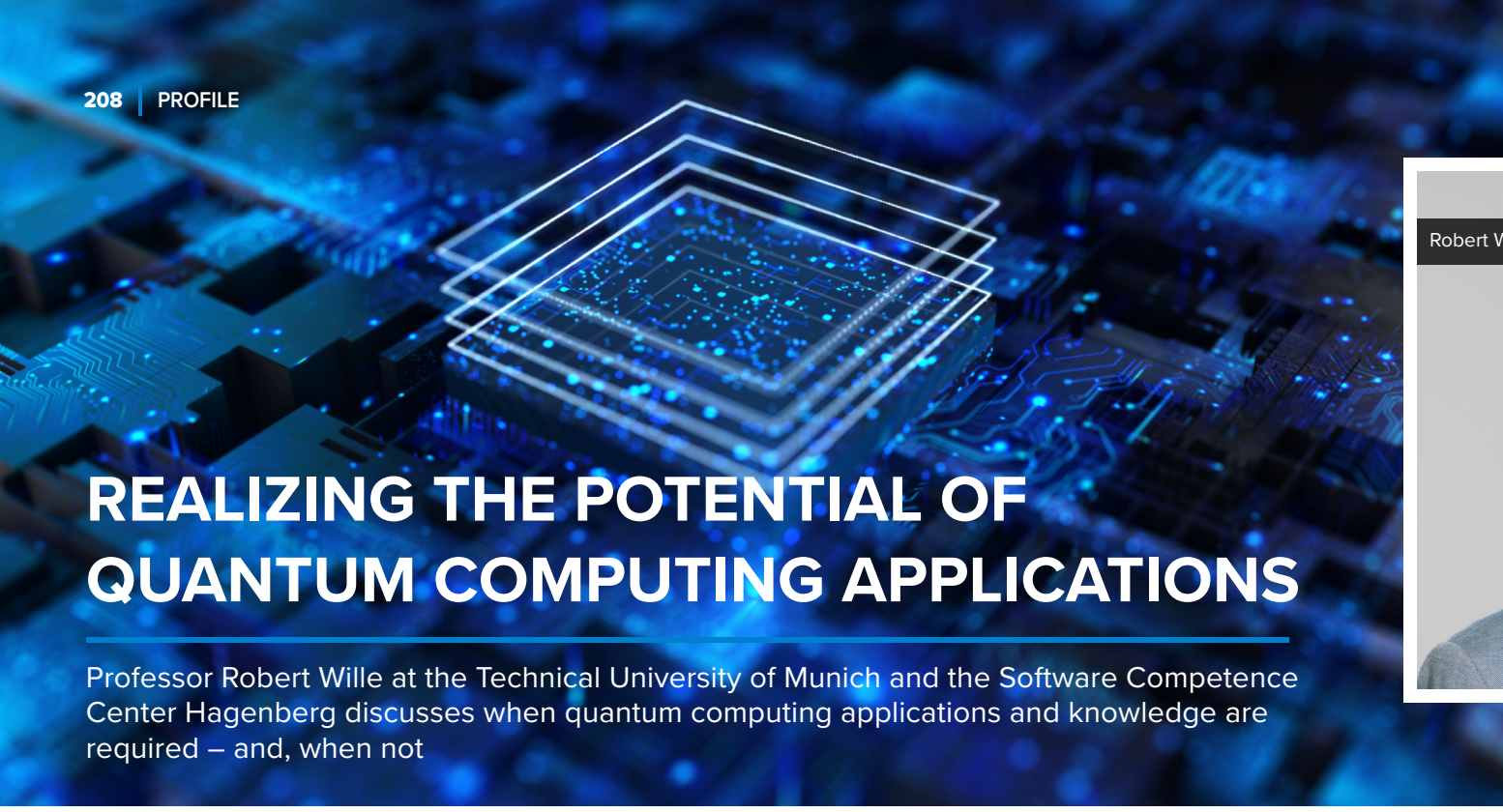

REALIZING THE POTENTIAL OF QUANTUM COMPUTING APPLICATIONS

Professor Robert Wille at the Technical University of Munich and the Software Competence Center Hagenberg discusses when quantum computing applications and knowledge are required – and, when not

Quantum computing is a rapidly developing research domain with enormous progress in hardware and software. This progress sparks huge interest in both academia and industry – leading to many applications from various domains, such as physics, chemistry, finance, and optimization.

However, utilizing quantum computing applications is a challenging task that obviously requires expertise in this new computing paradigm. At the same time, we do not have to completely reinvent the wheel and should rely on classical computing expertise whenever possible. This article reviews the steps to realize quantum computing applications and discusses, when dedicated expertise is needed and, when not.

An abstract summary of the corresponding steps is provided in Figure 1 and dicussed in the following.

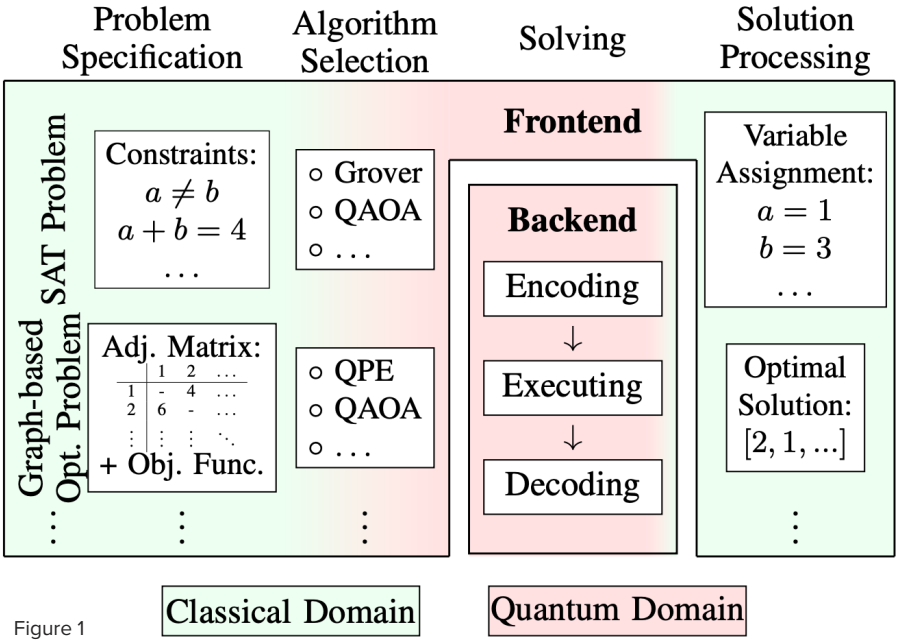

Figure 1

Problem specification

Many problems from various application and research domains can be clustered into problem classes with a shared problem description format. Examples are satisfiability (SAT) problems (which are defined by a list of constraints) or graph-based optimization problems (which are defined by an adjacency matrix and a respective objective function). These description formats are purely classical and are similar to the format conventional solutions expect.

Algorithm Selection

Currently, there are only a handful of quantum algorithms available which have shown to outperform their classical counterparts – either in theoretical complexity or in practical usage. Each of these quantum algorithms is solving a very specific class of problems and, therefore, expects the problem described in a corresponding respective format.

If a problem class can be formulated accordingly, the corresponding quantum algorithm is applicable. Examples are Grover's algorithm or QAOA for SAT problems and QPE or,


Ville

again, QAOA for graph-based optimization problems. Selecting a corresponding algorithm does not require in-depth expertise in quantum computing applications – a rough overview is sufficient.

To solve a problem from its respective problem class with a selected quantum algorithm involves three tasks which require expert quantum computing knowledge:

1. **Encoding:** The problem instance must be encoded as a quantum circuit such that the selected algorithm allows users to solve it.

2. **Executing:** The resulting quantum circuit must be executed on either a simulator or an actual quantum computer. For that, the quantum circuit needs to be compiled accordingly. The execution results in a histogram over computation outcomes.

3. **Decoding:** Exactly this histogram must now be decoded, such that the desired result can be extracted. The decoding scheme is dependent on the encoding scheme and may involve extensive postprocessing.

This step probably constitutes one of the biggest challenges for end-users who are "socialized" with classical computing and, now, want to utilize quantum computers. While this paradigm shift cannot be completely avoided, its ramifications can certainly be reduced by supporting the end-user with software and automation methods.

Solution Processing

The output of the previous steps provides the desired result of the originally given problem. As the final step, this result must be translated back into the original problem context and returned to the end-user. In the example of the SAT problem class, the variable assignment must be extracted from the result and represented in terms of a solution to the actual problem. In the example of the graph-based problem class, a sequence of graph nodes may be returned – depending on the objective function. Again, all these steps hardly require quantum computing expertise and mainly remain on the classical computing domain.

Whenever possible, end-users should be shielded from in-depth quantum computing aspects

Overall, most of the steps to realize quantum computing applications do not necessarily require quantum computing expertise. Considering that, thus far and in the near future, not everyone is going to become a quantum computing expert, this should be exploited when establishing corresponding design flows: Whenever possible, the end-users should be shielded from in-depth quantum computing aspects. This can be accomplished by proper (classical) interfaces and black box approaches – similar to conventional solutions (whose internal workings are often shielded from the end-user as well).

The quantum aspects which cannot be completely avoided obviously have to be handled by quantum computing experts. But even here, the threshold of required expertise can substantially be reduced by automation. Software and design automation methods as developed for the Munich Quantum Toolkit (MQT) aim at that by providing push-button solutions aiding the end-user, e.g., in automatically determining the most suitable platform, compiler, etc., offering simulation methods, compiler optimizations, and more. Those tools are publicly available as open-source projects on GitHub.

Interested in more?

Hungry for more detail? Check out the following articles, which provide more in-depth information, explicit examples, and references for further reading:

1. N. Quetschlich, L. Burgholzer, and R. Wille. Predicting Good Quantum Circuit Compilation Options. 2022. https://arxiv.org/abs/2210.08027

2. B. Poggel, N. Quetschlich, L. Burgholzer, R. Wille, and J. Lorenz. Recommending Solution Paths for Solving Optimization Problems with Quantum Computing. 2022. https://arxiv.org/abs/2212.11127

3. N. Quetschlich, L. Burgholzer, and R. Wille. Towards an Automated Framework for Realizing Quantum Computing Solutions. In International Symposium on Multiple-Valued Logic (ISMVL). 2023.

4. N. Quetschlich, L. Burgholzer, and R. Wille. Compiler Optimization for Quantum Computing Using Reinforcement Learning. In Design Automation Conference (DAC). 2023.

MUNICH QUANTUM TOOLKIT

Robert Wille
Professor
Technical University of Munich and the Software Competence Center Hagenberg
Tel: +49 176 23 44 09 64
mail@rwille.de
www.rwille.de