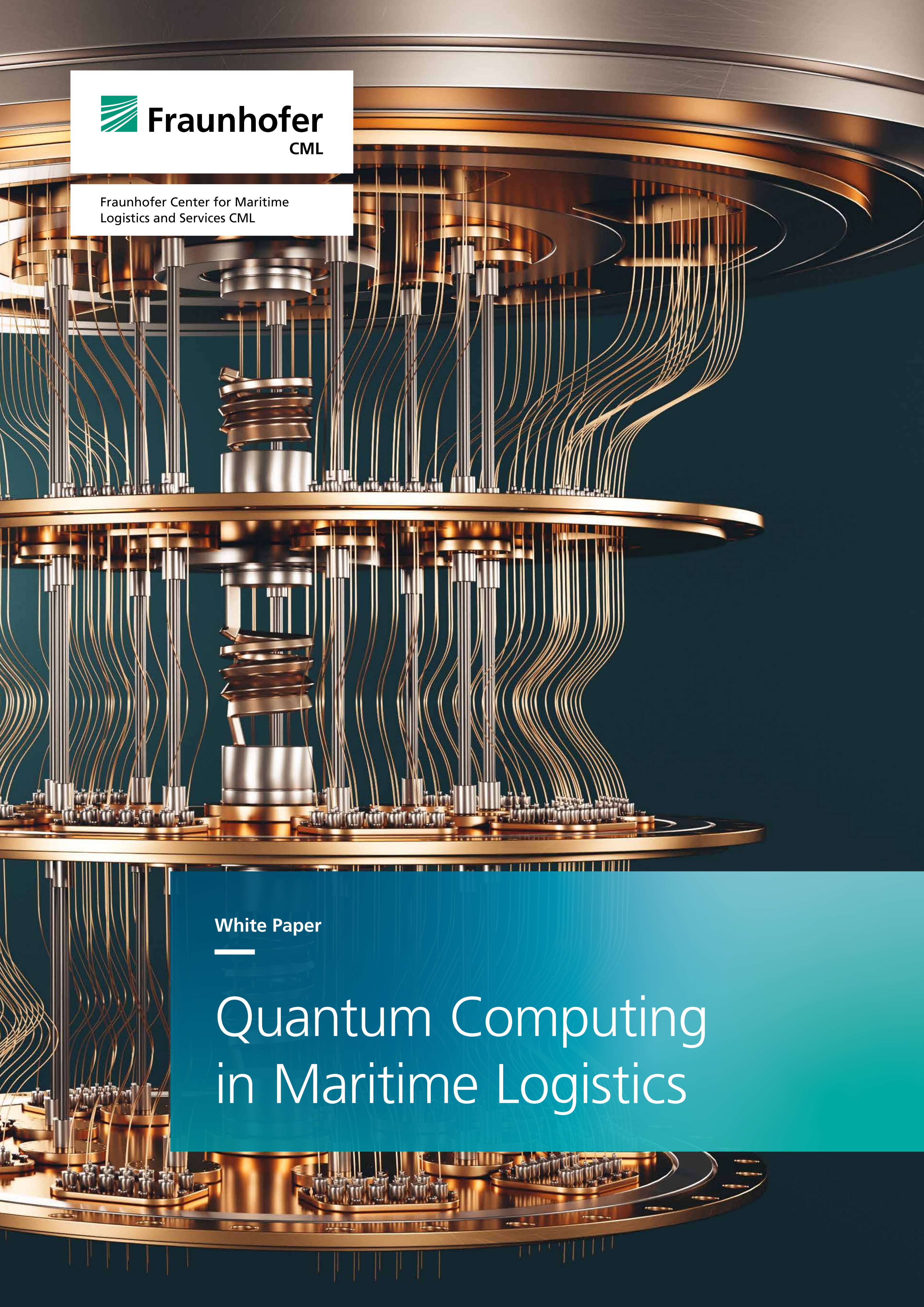


Fraunhofer
CML

Fraunhofer Center for Maritime Logistics and Services CML

White Paper

Quantum Computing in Maritime Logistics

Foreword

Dear Reader,

quantum computing is shaping up to revolutionize scientific computing. It has great potential in facing the current resource challenges, among others, in chemical simulations, machine learning, or combinatorial optimization in logistics. Over the last decade, its research has evolved from university labs trying to implement single qubits, the minimal building block for a quantum computer, to large companies as well as start-ups offering publicly available early-stage quantum computers as a cloud services. With the number of qubits, that are available in these cloud quantum computers, recently having doubled every year and the accuracy of these devices steadily growing, this revolution is only a matter of time.

Quantum computing achieves its speedup over classical computers through algorithms, that utilize quantum mechanical effects. Therefore, developing quantum computing applications is more complex than just replacing a classical computer with a quantum computer. It requires carefully chosen applications and specific algorithms.

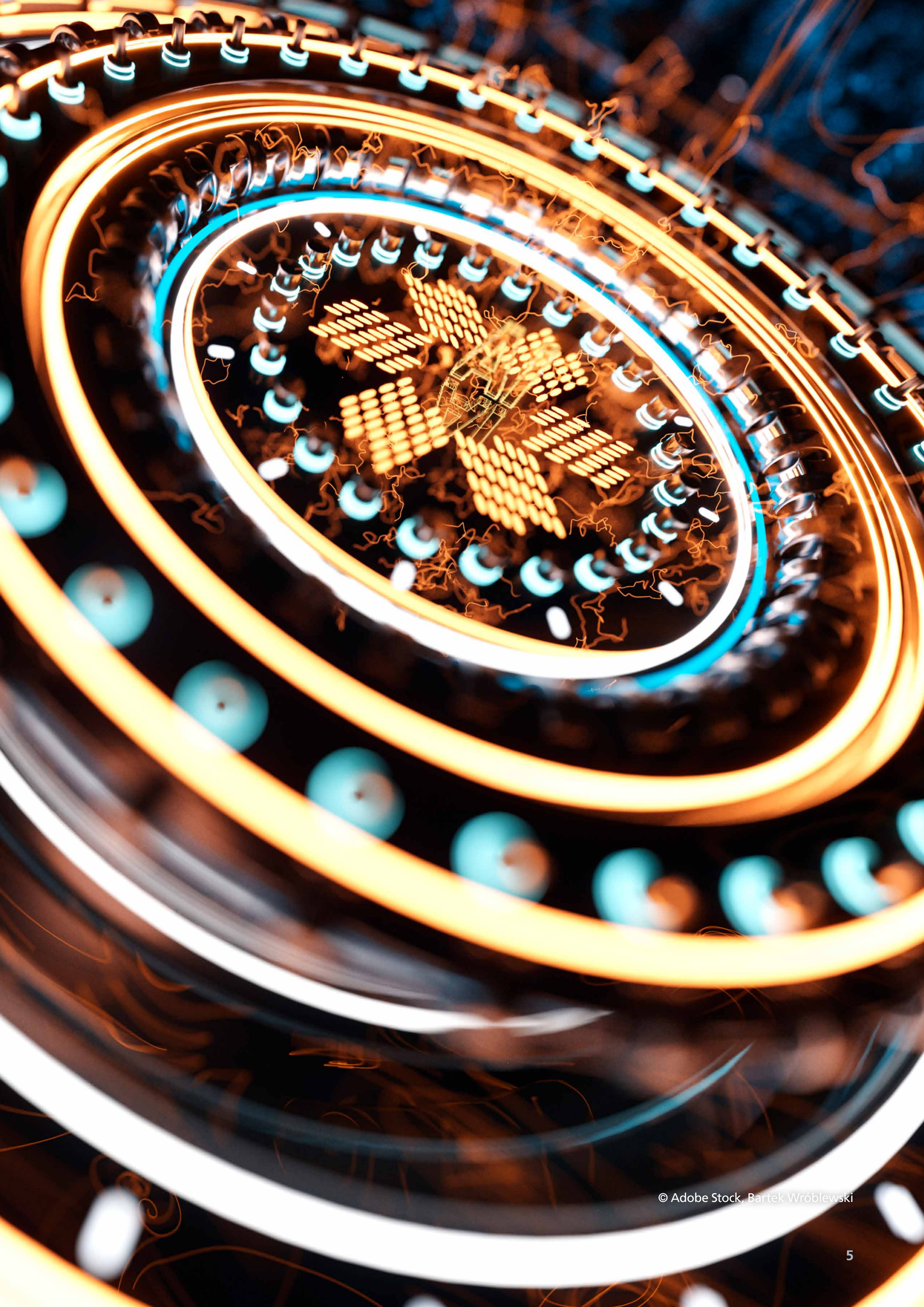
In maritime logistics, these applications could be combinatorial optimization problems, such as routing, network optimization, crew scheduling, or stowage planning, where the quantum computer acts as an enabler to solve more complex problems than ever, or massively accelerate existing calculations. While the development of quantum computers themselves are done by specialized companies, the development of applications in logistics is only possible in cooperation with the problem owner and business stakeholder.

The Fraunhofer CML combines knowledge and experience in maritime logistics, mathematical optimization, and quantum computing, making it the perfect candidate to support businesses in adopting this new technology.

I hope you enjoy reading this white paper on quantum computing, its potentials and challenges!

Prof. Dr.-Ing. Carlos Jahn
Director Fraunhofer CML

Content




© Adobe Stock, zhu difeng

1. Executive Summary

Quantum computing (QC) hardware is rapidly developing. If scaling and improvements in the logical accuracy keep developing at the current pace, we should soon see quantum computers (QCs) making their way into business applications.

QCs have the potential for a game-changing effect on combinatorial optimization. In maritime logistics, this could massively improve planning processes, such as crew scheduling, stowage planning, routing, or network optimization.

To get ready for utilizing QCs as soon as possible, it is now time to start preparing! In our opinion, future users need to start with two things:

- Use case identification and development: Quantum computing promises solutions for harder problems than ever before. To achieve that, the first step is to find answers to the question, how individual businesses can benefit from this improvement in mathematical capabilities and gain a real-world competitive advantage.
- Know-how acquisition and algorithm development: The different computing model of QCs as compared to classical computers calls for innovative algorithms. QC algorithms exploit the unique features of the computing model, which algorithms for classical calculations would not be capable of.

Within the scope of R&D projects, Fraunhofer CML supports businesses from maritime logistics in taking exactly those steps with a multi-platform approach. By including bridge technologies and QCs, we actively drive the development of the use case, while also developing specialized quantum algorithms on downsized proof of principle quantum calculations. This allows to migrate the calculation to future QCs as soon as they are technically mature enough to outperform their classical counterparts. Through close collaboration with our partners, we also support the acquisition of know-how on QC with a hands-on approach.

2. Quantum Computing and Maritime Logistics

Maritime logistics are the backbone of worldwide trade. Various actors, for example shipping companies, container depots, ports as well as logistic companies are responsible for managing and delivering huge amounts of goods, on time and to the right places. To achieve this, logistics companies need to face and efficiently solve various complex challenges with multiple conflicting requirements. For example, planning:

- the order of ports to be visited by a ship, taking cargo, customer, and port specific demands into account,
- the most cost-efficient route for an empty container to be shipped to the right place at the right time, considering different transportation modes, with relevant real-world limits and requirements,
- the best long-term crew assignment to a large fleet of ships, while complying with various legislative and company-specific regulations,
- the optimal schedule of tugboats for arriving and departing vessels,
- efficient crane movements or routes for vehicles in a port to deliver on-time service in a dynamic environment with uncertainties.

All these problems form the core of the businesses in maritime logistics and are characterized by a huge number of possibilities, which makes finding the best one a hard task. Naturally, companies are highly interested in adequate methods for solving such combinatorial optimization problems and thereby improve costs, quality, and environmental impact. To achieve this, two main challenges arise when applying optimization in the operative business:

1. In a dynamical environment, outside conditions tend to change. Therefore, an adequate formalization of the problem requires domain knowledge and foresight.
2. Combinatorial problems arising in practice are among the most complex optimization problems. Even high-performance computers often reach their limits when tackling medium-sized real-world instances.

Quantum and quantum-inspired computing promise to have a game-changing effect on mathematical optimization [1]. Due to their fundamentally different calculational model, they can solve certain problems using significantly fewer computational resources e.g., logical operations. When applied adequately,


© Adobe Stock, Travel mania

this technology can provide valuable services in maritime logistics.

State-of-the-art quantum computers (QCs) are closing in on the technical maturity required to outperform classical computers in solving practical problems. Over the last couple of years, their number of qubits, a quantum data storage unit, has been rapidly growing and the error rates in their logical operations are dropping. With those developments, economic, rather than purely scientific use cases seem reachable within the next few years. This prospect has sparked strongly increased interest in the topic. Investments, as well as public interest, in this technology are rising massively and commercial companies are at the forefront of the technical development. Some of the most prominent companies working on the commercialization of quantum computing are: Rigetti [2], Honeywell [3], D-Wave [4], IonQ [5], Google [6], and IBM [7]. With commercial quantum computing on the horizon, it is time for maritime logistics to start building a readiness plan for this upcoming technology disruption.

This white paper starts by introducing one of the biggest problems for mathematical optimization in logistics, namely the exploding requirement of calculational resources, in problems of realistic size. It then goes on, by explaining the potential of quantum computing with respect to this problem. The next section first briefly discusses the capabilities of current QCs and their development. This is used to set up a roadmap to support businesses in getting ready for quantum computing. Last, we show possibilities for collaboration with the Fraunhofer CML.

3. Combinatorial Explosion and Limits of Computational Resources

Due to ever cheaper and more powerful computers available to everyone nowadays, computational resources often seem practically infinite to average users. However, in some areas, such as scientific computing, optimization, cryptography, or artificial intelligence, this is not the case. Here, problems arise which are so complex, that solving them demands unrealistic amounts of calculational power. Real-world maritime logistics problems tend to suffer from that.

In combinatorial optimization problems, the number of possibilities increases drastically with the problem size. To understand this better, let's consider an example: a simplified variant of fleetwide voyage planning in shipping companies (Fig. 1). In a generic form, this problem is defined by a set of delivery jobs, which have a pick-up harbor, a drop-off harbor, and a capacity requirement for the freight (usually weight or volume), as well as a set of ships with maximum loading capacities. The task is to plan routes for each ship, such that all delivery jobs are

Fleetwide Voyage Planning

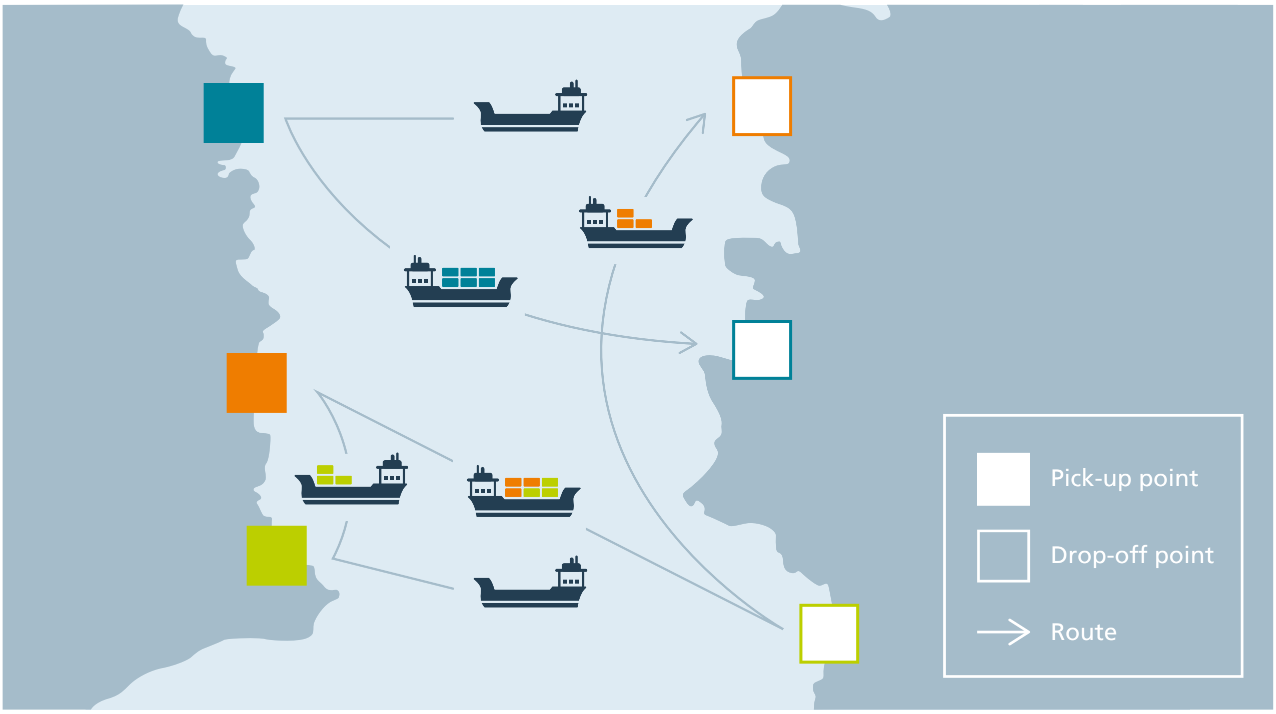


Fig. 1: Illustration of a fleetwide voyage planning problem. Delivery jobs need to be assigned to ships and their routes must be planned. Additionally, the ships may not be overloaded at any point.

processed while complying with the maximum loading capacity of every ship.

The number of possible routes, from which we need to pick the best one, scales strongly with the number of harbors (or jobs). In other words, if the number of harbors is increased slightly, the number of possibilities grows tremendously. Of course, in such routing problems, the number of possibilities depends on the details of the application, for example further constraints or the number of ships in use, but characteristically for n harbors the number of possibilities grows roughly as $n!=1*2*\ldots*(n-1)*n$. The fact, that every additional harbor increases the number of possibilities by a factor, rather than an absolute amount, causes the number of possibilities to quickly grow to amounts that cannot be handled with any kind of computer. To give an idea how severe such scaling problems are, here is a placative example:

60 harbors already allow for more possible routes than there are particles in the universe (roughly 10^{80}).

The above introduced and simplified example of fleetwide voyage planning illustrated the problem of combinatorial optimization. Typically, there are too many possibilities to individually search through them all, regardless of whether we use a laptop or a supercomputer. Such problems are usually tackled by heuristic methods. They make application-specific assumptions/approximations about the optimal solution and use that to guide the search. While this guided search can massively reduce the required calculation time to find a good solution, they usually cannot guarantee the optimality of the solution. Afterall, a guided search is merely an educated guess on which parts of the search space are most likely to include the optimal solution. So, with heuristics, faster calculation times do imply that the search space has not been fully checked. Since a reduction of the search space also reduces the chance of including the best solutions, heuristics introduce a tradeoff between the complexity of the model to solve, the required calculation time and the quality of the solution (Fig. 2).

More powerful computers require fewer compromises between the calculation time and the quality of the result. This could improve business applications, because higher quality solutions translate to more efficient allocation of business resources, while lower calculation times allow easier replanning, i.e., higher flexibility in dynamical surroundings.

Heuristics

Fig. 2: Heuristics approximate complex calculation tasks to simplify them. The freedom to introduce more, or cruder approximations creates a tradeoff between the required calculation time, complexity of the optimization task and the quality of the solution.

4. Quantum Computing: Introduction and Potential

Scaling of required computational resources is central to quantum computing. While classical computers have been developed to have ever-growing memories and frequencies of logical operations, the big promise of quantum computing is not to increase this even further. Instead, QCs calculate more efficiently due to their totally different way of working. More precisely, there are certain problems, where the calculational resources (especially time) a QC requires scale less harsh than those a classical computer requires. Some of the major fields in which QCs offer great potential are listed in Fig. 3.

Quantum computers can achieve this improved scaling of computational resources by offering different resources combined with different algorithms. Instead of normal bits, they have qubits, which do not only allow values of zero or one, but also so-called quantum superpositions. To understand this distinction, one should think of the values, that a qubit can take, as possibilities, with probabilities assigned to each of them. This is fundamentally different, because a classical bit always has a deterministic state, i.e., cannot have statistical states. A qubit on the other hand, could, for example, be in a 50:50 state, in which both values are equally likely (Fig. 4).

Potential Quantum Computing Applications

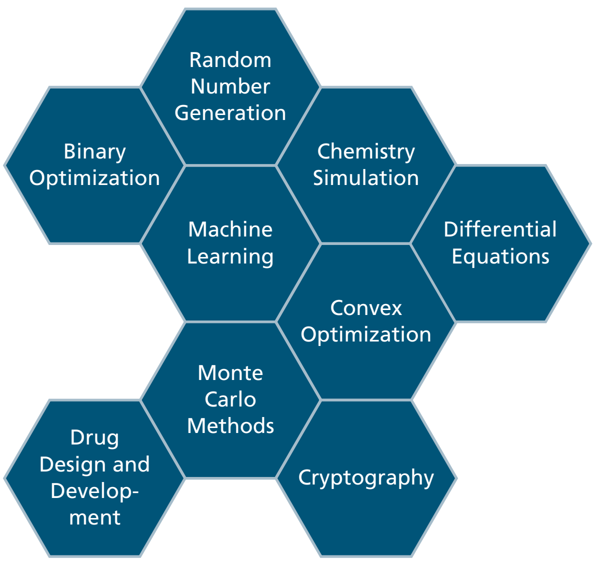


Fig. 3: Quantum Computing will have a major impact in many fields. Its potential in optimization and related applications make it interesting for logistics.

Bits vs. Qubits

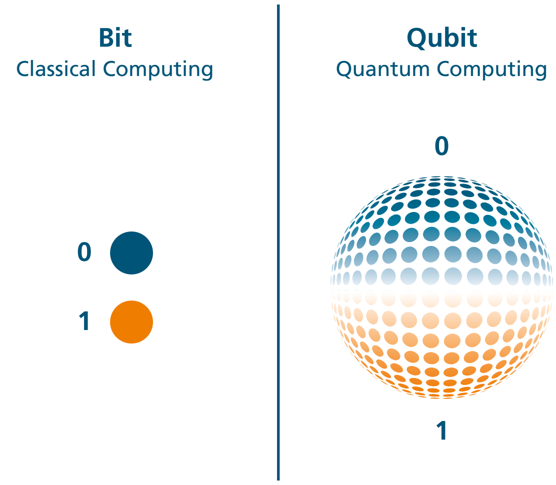


Fig. 4: While classical bits can only have one of two well-defined states, a Qubit can also be in statistical mixtures of those two classical states. This forms a continuous state space which is typically visualized as a sphere, where the longitude translates into probabilities for either classical state. Here these probabilities are color coded.

This seemingly small difference between classical bits and qubits has huge implications. While *n* classical bits can take 2^n different states, *n* qbits can take any quantum superposition, i.e., statistical mixture of these states, thereby forming a 2^n dimensional space, also known as vector space. While the technicalities of this mathematical construct are not the topic of this work, it is important to understand, that this kind of vector space cannot easily be simulated on a classical computer. In fact, it would have the same issue as other complex problems, for example the fleetwide voyage planning problem introduced in the last section: it scales badly. Simulating one additional qubit doubles the problem size. In a QC, this huge state space does not have to be simulated though, but it is physically implemented, and we can operate on it easily and efficiently. So, in essence, quantum computing utilizes a harshly scaling state space for solving harshly scaling problems to outperform classical computers.

One less abstract way to illustrate the advantages of those probabilistic states is the idea of quantum parallelism. A quantum computer can apply a function to a superposition (statistical mixture) of inputs, which will result in a superposition of results. A classical computer cannot do that. To apply a function to multiple inputs, it needs to either work sequentially or, it needs multiple processors for normal parallelization. The shortcoming of simply imagining QC as a way of parallelization is, that the result is also a quantum superposition. Extracting a meaningful answer, i.e., a concrete non-statistical result, from that superposition is often the main problem in quantum algorithms (for example so-called amplitude amplification).

While the high dimensional state space of a quantum computer is immensely powerful, it is not always trivial to utilize it and gain an advantage over classical algorithms. Examples of problems and corresponding algorithms, that offer a scaling improvement over classical calculations do exist though. The most famous one is probably Shor's algorithm for prime factorization, which yields an exponential speed up [8] [9]. In other words, the calculation time required to solve this task on a classical computer is an exponential of the length of the problem. On a QC, the calculation time is a polynomial of the length of the input data, so the quantum speedup grows with the problem size. However, the potential application of Shor's algorithm is in cryptography and therefore, not directly relevant in logistics.

For logistics, we are interested in optimization and, related to it, search algorithms. Grover's algorithm is a search algorithm that searches through *N* possibilities, but only scales with $\sqrt{N}$ [10]. That means for a sufficiently large search space, it requires less steps than there are possibilities, even though it does check every single one (Fig. 5)! This kind of search is strongly related to optimization – Grover's algorithm searches an item that fulfils certain conditions, an optimization algorithm searches an item that optimizes a function. Both tasks become difficult, if the number of possibilities is very large. Because of the similarity of the problems, there is hope to speedup optimization in a similar fashion as normal searches, for example by using the Quantum Approximate Optimization Algorithm (QAOA), a quantum optimization algorithm for generic functions. But even if QAOA cannot live up to those expectations, we could still decompose optimization tasks into multiple decision (i.e., search) problems and thereby directly utilize the speedup of Grover's algorithm in optimization tasks.

The examples above clearly show one important fact: the speed-up, that a quantum computer offers, depends strongly on the problem one wants to solve. It does not automatically outperform classical computers on everything. Quite the opposite, classical computers are cheaper, larger, simpler and, at least at this point of technical development, more reliable. Therefore, quantum computers are only useful if one leverages their advantages.

Due to this tradeoff between scaling advantages and disadvantages in size and cost of the system itself, the choice of use cases is essential to applied quantum computing. The perspective QCs offer for optimization are scaling improvements in combinatorial optimization. Such improvements can either enable us to solve problems that were not feasible before, allow us to improve the approximations and thereby increase the quality of the solution, or simply reduce calculation times to increase the flexibility in planning.

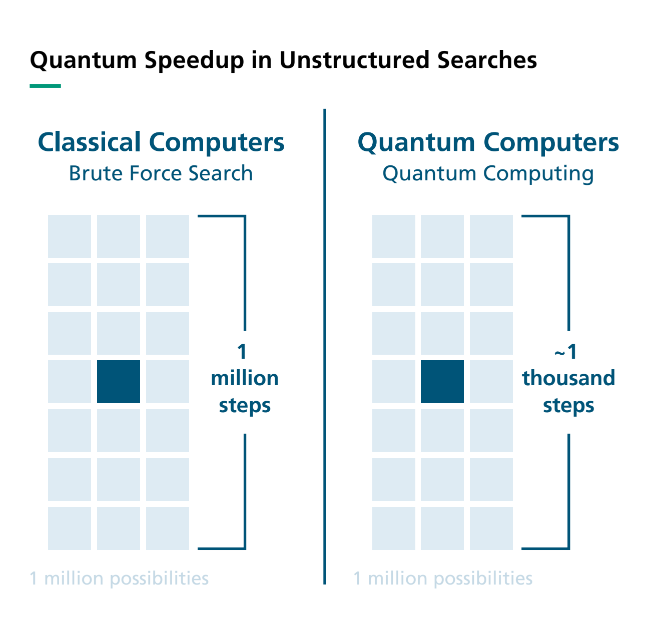


Fig. 5: When searching through a set of possibilities, classical computers must check every possibility individually, therefore the calculation time grows linearly with the number of possibilities. Quantum Computers can accelerate this process and the required calculation time then only grows with the square root of the number of possibilities

5. Current state of Technology

5.1 QC Hardware

There has been remarkable progress in the development of quantum computers over the last couple of years. In contrast to classical computers, where there is one well-established hardware technology, namely semiconducting transistors based on silicon, in quantum computing there is not yet one clear winner in the competition for the best quantum computing hardware platform.

In fact, there are not only different approaches to the physical implementation of qubits, but even multiple approaches to the calculational model. To name the most prominent ones:

- gate-based universal quantum computing
- measurement-based quantum computing
- annealing.

Even though there are vastly different approaches to building QCs, the fundamental problems are the same: First, scaling the size of a quantum computer is much harder than scaling a classical system. Second, as opposed to digital computers, QCs, as probabilistic machines, have a continuous state space which makes them vulnerable to arbitrarily small sources of errors. To understand this better, let's start with classical bits: they can only be in one of two clearly distinct states, 0 or 1, and switching between those two sates requires a control pulse of some minimum strength. If any weak perturbation from the surroundings hits the classical bit, nothing happens. This results in classical bits, that work almost without any errors. A quantum bit on the other hand, has a continuous state space, i.e., it is possible to make arbitrarily small changes to the state of the qubit. The problem is that arbitrarily small perturbations from the surroundings then induce (small) errors to the state of the qubit. Even though the single errors might be small, they add up, thereby quickly destroying the data encoded in the qubit. This is actually also the reason why most quantum computers are cooled down to nearly zero temperature, because it reduces the amount of thermal perturbations in the system.

Both these issues confine the size of feasible quantum calculations. While the size of the device limits the amount of data, that can be stored and processed, the propensity to errors limits the length of the calculation. If a program is so long, that statistically logical errors are expected to occur, the result is not trustworthy. In principle that also applies to classical computers. Practically it is not an issue though, because their error rates are just low enough (roughly a factor 10^{24} smaller than the most advanced quantum computers).

The largest available (universal) quantum computers just recently broke the 50 and 100-qubit barriers [6] [11] [7]. While that is an impressive technological achievement, it still means, that current quantum computers can only work on problems with variables that fit in roughly 100 bits. Of course, most real-world problems are larger than that (at least in logistics).

In these QCs, the chance for errors during any calculation, is still too high to do useful calculations, even ignoring size constrictions. In the long run, this shortcoming is not as severe as it might seem though, because of so-called quantum error correction. It describes the possibility to bundle multiple physical qubits together into one logical qubit, thereby creating redundancy, and to actively correct errors in the physical qubits [12]. Implementing quantum error correction is technically demanding, because it requires a relatively high minimal accuracy in the underlying physical qubits. There are different estimates for this requirement, ranging from an error rate of roughly 10^{-2} to well below 10^{-3} (errors per operation), due to differences in the assumed noise model and error correction technique.

Since error correction performs better with more physical qubits per logical qubit, it introduces a tradeoff between accuracy and qubit number. This grants the ability to build arbitrarily reliable qubits and would be a huge milestone for quantum computing, so there has been a lot of work towards demonstrating it in experiments, for example [13] [14] [15] [16]. With error rates of less than 10^{-3} even for 2-qbit operation, current technology is approaching this milestone.

At first sight, the improvement of accuracy at the cost of reducing the size of quantum computers might not seem to be very useful, because right now size is already lacking. The important thing to notice is the speedup in increasing the size of quantum computers. This speedup is most visible in the developments of the last few years and the developments to come. While it took many years to develop decent qubits and not so long ago the step from one qubit to 3 or 5 qubits on one chip was a significant technical challenge, now there are plans to increase the qubit number from roughly 100 to over 1000 within only two years [17]. While those plans are ambitious, they do seem realistic, because the most fundamental scaling issues already had to be solved for the current generation of devices. Now the improvements are more continuous resulting in continuously increasing sizes of QCs (Fig. 6).

Quantum Computing Timeline

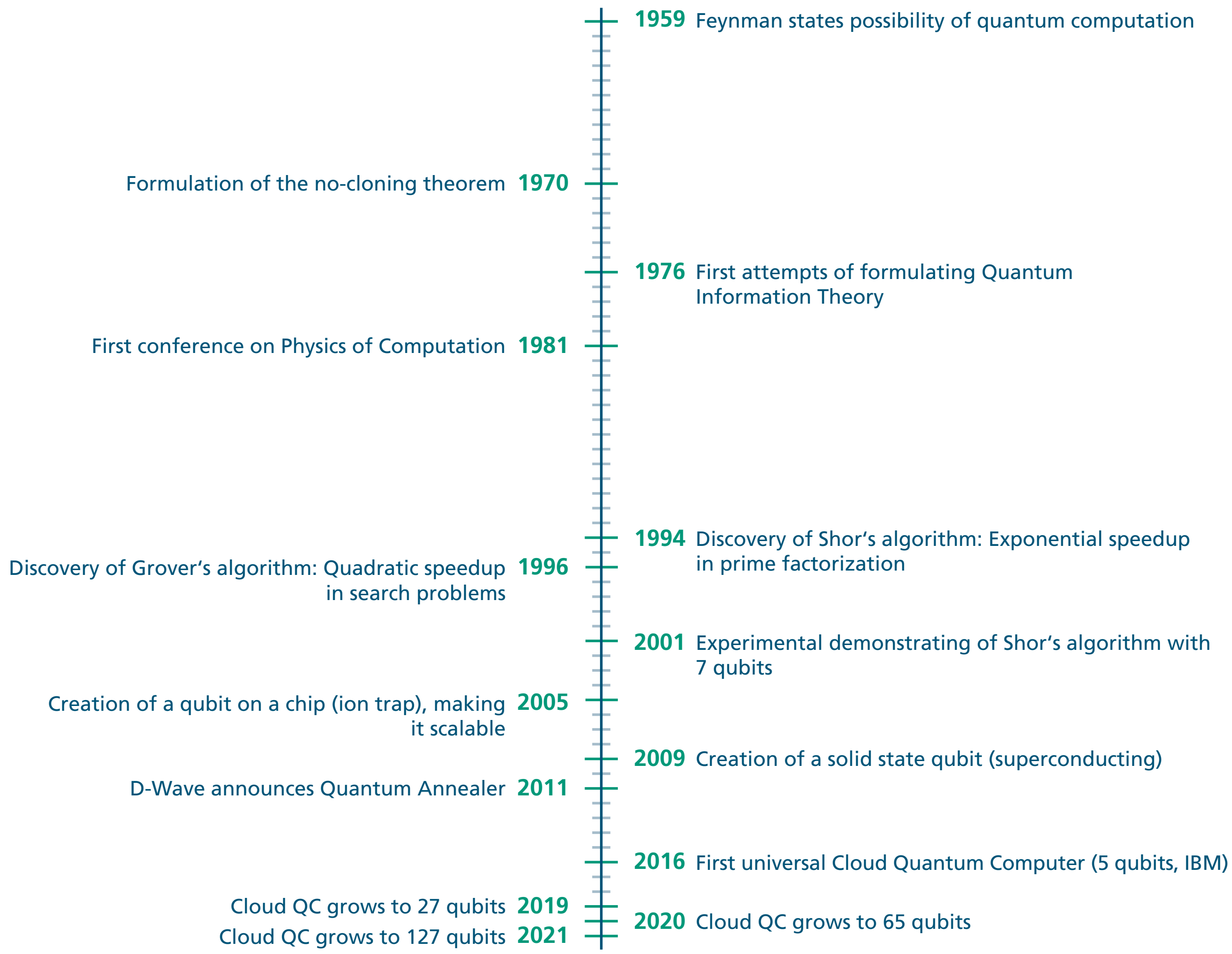


Fig. 6: The development of QCs has drastically accelerated. In addition, the research objective has shifted. The groundwork is established, now there is a race for the best technical realization of QCs.

5.2 QC Algorithms

To utilize QC, we do not only need good hardware, but also algorithms that solve real-world problems while using the advantages of a QC. We already pointed to scaling speedups a QC can achieve over a classical computer. As mentioned before, in the context of maritime logistics, we mostly care about optimization and possibly search problems, hoping for a quadratic speedup. So, just like the Grover algorithm can search through n^2 possibilities in ~n steps, we hope for the same effect in QAOA.

While a quadratic speedup would be a great technological achievement, there is one problem though. Many problems, in logistics, scale exponentially. Since the square root of an exponential is still an exponential, the scaling would not be qualitatively improved, instead only the exponent would be divided by two, thereby only doubling the size of feasible problems. It seems like a rather specific case, in which doubling the feasible problem size is worth the cost of using QCs.

Especially in logistics, most planning tasks are just too large if the underlying calculation scales exponentially (regardless of a factor 2 in the exponent). Therefore, we believe, that in order to utilize quantum computing in practice, we need to do the same thing as in classical computing, namely develop problem-dependent heuristics, that yield a tradeoff between required calculational resources and the quality of the solution. If we succeed in combining a heuristic that scales polynomial with a quadratic speedup, this will result in a qualitative scaling improvement, namely a lower polynomial order.

A major challenge to this approach is the difficulty to design algorithms for QCs, which offer a scaling speedup over classical systems. But as a first step, designing completely new algorithms might not even be necessary. Instead using existing algorithms, like the Grover algorithm or QAOA, and developing heuristics that bias the search space already has the potential to outperform classical computers, because it combines the speedup from using a heuristic with the QC's scaling speedup. Technically it is already known how to bias the search space [18] [19] [20], namely by adapting the so-called mixer, an operator that is part of the algorithms [21]. Therefore, the next step is to develop suitable heuristics for specific logistic applications, implement them and test them.

Developing applications for QCs, including the identification of suitable problems, creating mathematical models, and customizing quantum algorithms, is hindered by the currently immature technology. Developing the use case and its mathematical model requires calculations with realistic, i.e., large data instances – realistic problems tend not to fit in roughly 100 bits - but developing quantum algorithms requires experiments with a QC or simulation environment, which are small and in the case of real QCs prone to error. To parallelize the development of the use case and the algorithm, we plan to include QCs and bridge technologies in our work.

5.3 Computing Platforms

There are various bridge technologies between normal classical computers and QCs ranging from simulation environments running on GPU clusters to so-called annealers, which are specialized computers for one form of optimization problem, namely quadratic unconstrained binary optimization (QUBO) tasks. This confinement allows annealers to be specifically designed for one algorithm, thereby significantly boosting their performance. Let's first get an overview over the computing platforms that we consider for our work:

- Digital annealer (for example Fujitsu, Hitachi): A classical annealer based on bits, not qubits. As a classical system, it is technically mature, i.e., works reliably and at useful scales (~8000 bit). It uses thermal annealing, a well understood process, as its optimization algorithm and has a chip

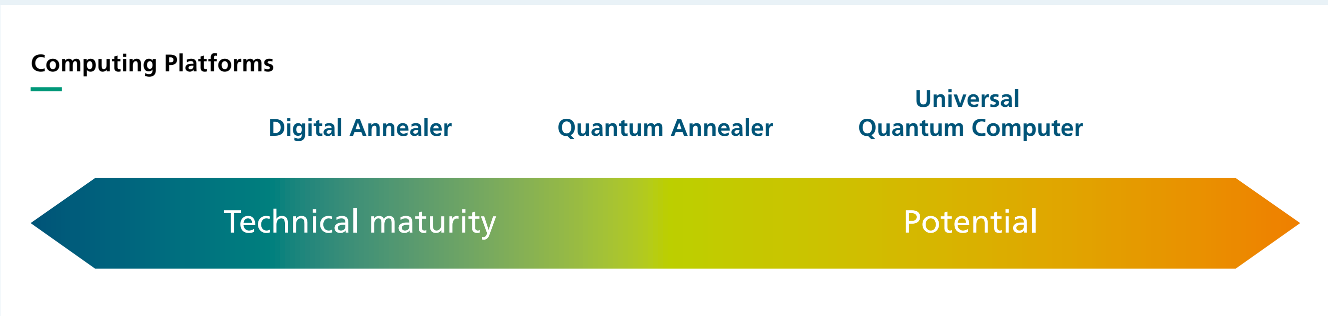
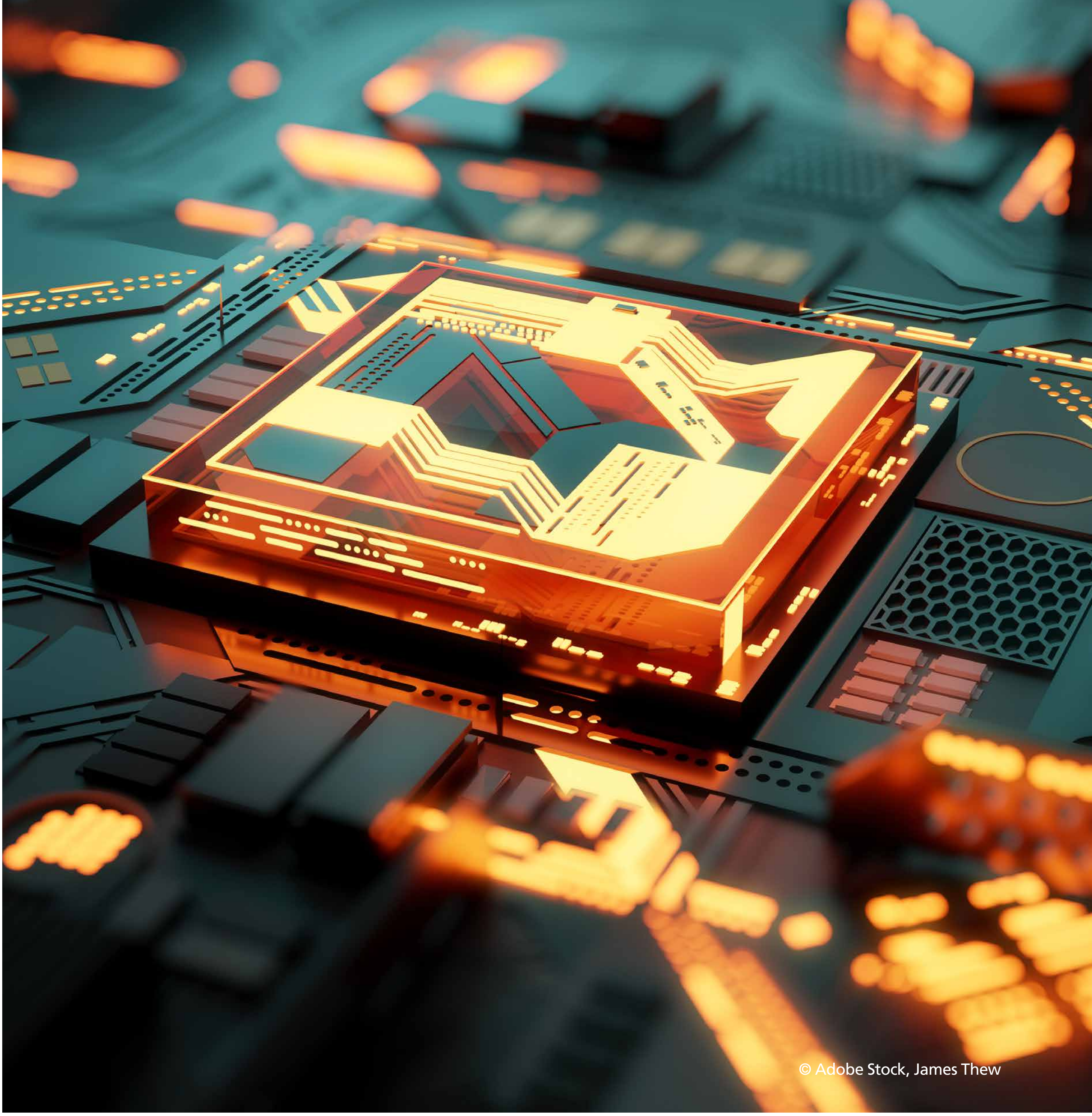


Fig. 7: Estimation of both, the potential, and the technical maturity of the computing platforms we want to use. At this point, there is a clear tradeoff between them.


© Adobe Stock, James Thew

architecture that is optimized for exactly this algorithm. On the downside, as a classical device, it does not offer the potential for a scaling speedup like quantum devices.

- Quantum annealer (D-wave): The quantum analogue to the digital annealer, based on qubits and using quantum annealing as its algorithm. Due to its concept as a special purpose machine, its technical requirements are lower than that of a universal quantum computer. That translates to a greater technical maturity, which is most visible through the scale of these systems (~5000 qubits). On the downside, the question whether these devices, with their current hardware and algorithm, offer a scaling speedup over classical systems is subject to controversy. It would be interesting to let a digital annealer and a quantum annealer compete on some applied problems.
- Simulation environment: It is possible to simulate sufficiently small quantum calculations on classical computers. This allows the development of algorithms in a controlled environment, which mainly refers to error free simulated qubits, or controlled error models. This avoids the implicit length restriction on quantum algorithms for current QCs.
- Universal quantum computers: As discussed, these devices are not yet technically mature, but offer the potential for scaling improvements. As an advantage over the quantum annealer, it offers the possibility to run arbitrary algorithms and therefore solve arbitrary problems.

6. What now? Developing QC Applications

Quantum Computing is a technology with great potential, but still with insufficient technical maturity. The potential for drastic improvements in calculation speed, to a point that it will enable completely new use cases for mathematical optimization in maritime logistics, is therefore concealed by the current state of hardware technology. This makes the development of QC applications challenging, because testing mathematical models and algorithms is not straight forward. After all, the device these highly complex calculations should run on, is not yet ready.

Especially the identification of use cases and developing suitable mathematical models, that accurately represent the real-world challenges faced in the maritime industry, should not be underestimated. In opposition to that, discussions about QCs usually focus on solving mathematical models but not on defining them. This highlights their role as enablers but does not work towards providing full solutions to problems in logistics. To allow testing and iterative development of applications and mathematical models, one needs to be able to solve realistically sized optimization problems. This is possible by confining the form of the

Quantum Roadmap

Platform-Independent Modeling

- Identify suitable use case
- Develop mathematical model
- Express as QUBO
- Develop frontend:
  – Tranlation to QUBO parameter
  – Translation of QUBO solution

Backend Solver

- Choose calculational platforms
- Create downsized problem for QC
- Develop solver

Test & Evaluation

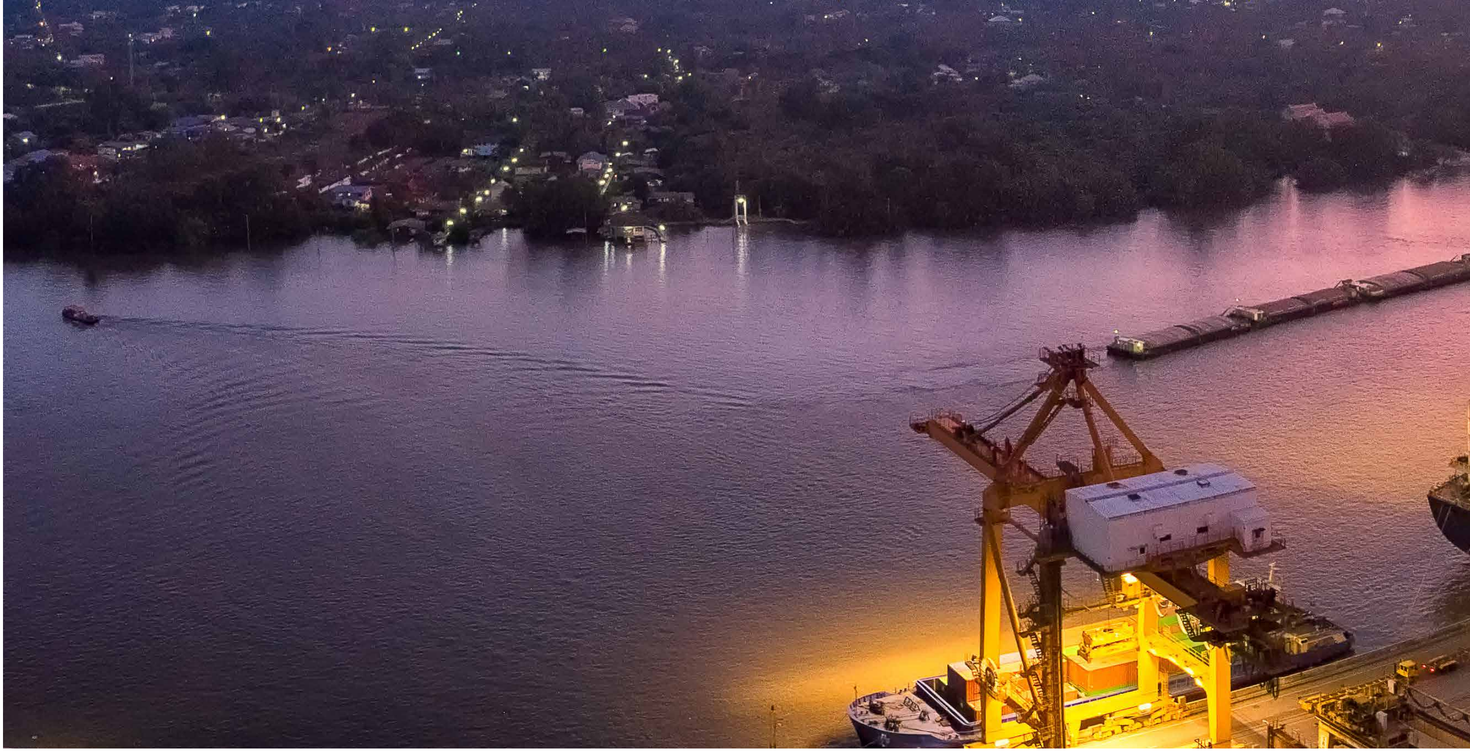
- Evaluate and adapt:
  – use case definition
  – mathematical model
  – optimization target
- Test solvers
- Evaluate short-, mid- and long-term potential of each platform

Fig. 8: Our roadmap for research projects. To iteratively develop the use case and the quantum algorithm, we include different computing platforms, while minimizing redundant work.

mathematical models that we develop to QUBO and using (digital and/or quantum) annealers. Annealers seem like a great choice, because QC applications will always be complex problems, such that large amounts of calculation power are necessary. Of course, they do not offer the performance we expect from a mature QC, but since testing does not have the same time and quality constraints as an operational environment, they are sufficient for the development of QC applications.

While annealers allow the development of use cases, they are useless for the development of quantum algorithms for optimization. Developing those requires using current QCs and/or simulation environments, where downsized versions of the original optimization problem can be run. Including different computing platforms allows the parallelization of both developments.

Especially when using multiple platforms for calculations, we define work modules (Fig. 8), which clearly distinguish between generic and platform-specific developments. Since the computing platform is only the solver of the mathematical optimization problem, most developments turn out to be generic.



7. Opportunities for Cooperation Current state of Technology

Fraunhofer CML supports companies in closing the gap between cutting edge technology developments and roll out to the operative business. It combines experience and know-how of both, quantum computing and maritime logistics, making it the perfect partner for the development of early-stage QC applications. Like the modular setup of the quantum roadmap, we also modularize our professional services by defining multiple levels for each project:

Level 1: Getting Started – Design Thinking

Who:

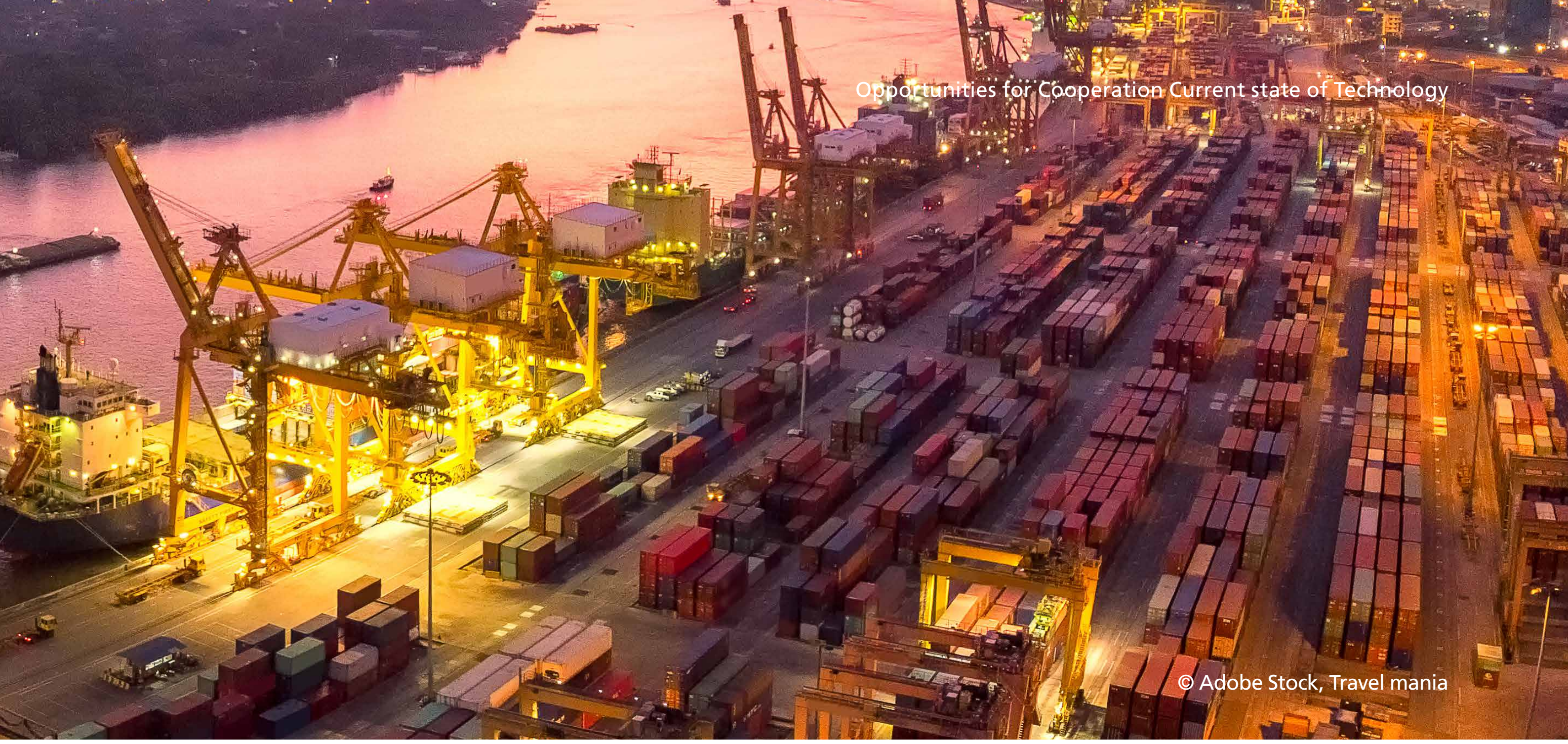
- Companies with limited experience/expertise in mathematical optimization or QC

What:

- Support in identifying potential for improvements in operational business through mathematical optimization
- First conceptualization of a software solution
- Feasibility study
- Benefit estimation of using QCs

Why:

- Low entrance barrier
- Focus on big picture and adding value
- Treat QC as a means not an end



© Adobe Stock, Travel mania

Level 2: Follow the Quantum Roadmap – Calculate and Iterate

Requires:

- A suitable use case (compare level 1)

What:

- Develop user interface
- Development of mathematical model
- Proof of principle and test calculations

Why:

- Test and improve use case with real data instances
- Acquire know-how and hands-on experience on QCs
- Quantitative estimate of operational benefit based on test calculations

Level 3: Final Solution – Creating a Software Tool

Requires:

- Development of use case and mathematical model completed
- Clear decisions on user interface
- At least one feasible solver developed (compare level 2)

What:

- Development of a software solution for use in operative business

Why

- Use the technology in operational business

References

[1] Dalzell, Alexander M., et al. How many qubits are needed for quantum computational supremacy? Quantum. 11. 5 2020.

[2] Rigetti. [Online] 20. 12 2021. https://www.rigetti.com/what-we-build.

[3] J. M. Pino, et al. Demonstration of the trapped-ion quantum CCD computer architecture. Nature. 7. 4 2021.

[4] M. W. Johnson, et al. Quantum annealing with manufactured spins. Nature. 11. 5 2011.

[5] K. Wright, et al. Benchmarking an 11-qubit quantum computer. Nature Communications. 11 2019.

[6] Frank Arute, et al. Quantum supremacy using a programmable superconducting processor. Nature. 23. 10 2019.

[7] IBM. Quantum Summit. 16. 11 2021.

[8] Simon, Daniel R. On the Power of Quantum Computation. SIAM Journal on Computing. 1997.

[9] Shor, Peter W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Review. 1999.

[10] Grover, K. A fast quantum mechanical algorithm for database search. ACM Digital Library. 1996.

[11] Yulin Wu, et al. Strong Quantum Computational Advantage Using a Superconducting Quantum Processor. Physical Review Letters. 25. 10 2021.

[12] Terhal, Barbara M. Quantum error correction for quantum memories. Reviews of Modern Physics. 7. 4 2015.

[13] Jay M. Gambetta, Jerry M. Chow & Matthias Steffen. Building logical qubits in a superconducting quantum computing system. Nature, npj Quantum Information. 13. 1 2017.

[14] Maika Takita, et al. Experimental Demonstration of Fault-Tolerant State Preparation. Physical Review Letters. 31. 10 2017.

[15] D. Ristè, et al. Detecting bit-flip errors in a logical qubit using stabilizer measurements. Nature Communications. 29. 4 2015.

[16] J. Kelly, et al. Scalable in situ qubit calibration during repetitive error detection. Physical Review A. 26. 9 2016.

[17] IBM. [Online] 20. 12 2021. https://research.ibm.com/blog/quantum-development-roadmap.

[18] Stuart Hadfield, et al. From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. MDPI Open Access Journals. 12. 2 2019.

[19] Daniel J. Egger, Jakub Mareček, and Stefan Woerner. Warm-starting quantum optimization. Quantum. 17. 6 2021.

[20] Wim van Dam, et al. Quantum Optimization Heuristics with an Application to Knapsack Problems. 8 2021.

[21] IBM. Solving combinatorial optimization problems using QAOA. Qiskit Textbook.

[22] S. Debnath, et al. Demonstration of a small programmable quantum computer with atomic qubits. Nature. 3. 8 2016.

Impressum

Editor:
Prof. Dr.-Ing. Carlos Jahn
Director Fraunhofer CML
carlos.jahn@cml.fraunhofer.de

Authors:
Sebastian Rubbert Ph.D.
Research Associate
Expert for Quantum Computing
sebastian.rubbert@cml.fraunhofer.de

Dipl.-Ing. Anisa Rizvanolli
Head of Team Maritime Scientific Computing and Optimization
Expert in Mathematical Optimization
anisa.rizvanolli@cml.fraunhofer.de

White Paper

Quantum Computing in Maritime Logistics

Cover photo acknowledgment:

Contact:
Fraunhofer Center for
Maritime Logistics and Services CML
Am Schwarzenberg-Campus 4D
21073 Hamburg
Germany

+49 (0) 40 42878-4450
info@cml.fraunhofer.de
www.cml.fraunhofer.de

Contact

Fraunhofer Center for
Maritime Logistics and Services CML

Am Schwarzenberg-Campus 4
Gebäude D
21073 Hamburg
Germany

Tel.: +49 (0)40 / 42878-4450
info@cml.fraunhofer.de