Quantum Computing for Software Engineering

ANTONIO TROVATO

ACM Reference Format:
ANTONIO TROVATO. 2022. Quantum Computing for Software Engineering. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

GitHub repository: https://github.com/AntonioTrovato/Quantum-Computing-for-Software-Engineering

1 CONTEXT OF THE PROJECT

Nowadays digital technologies are pervading every aspect of our society. Organizations, companies and more in general everyone of us rely to some degree on software products of different kinds. Since we are entrusting our lives and business to software, concepts such as cybersecurity and cyberdefense are becoming increasingly important. In a world where large amounts of sensitive data, managed by different DBMS, are the core business of digital products, software vendors must be capable of protect any aspect of their product, in order to safeguard individual privacy and business critic information.

Nevertheless, this is a complex topic since it requires a clear understanding of all parts of an organization, like personal or resources and many others, often not available. For this reason, the Information Management System (ISMS) is an important management structure focused on information security within an organization. ISMS defines security policies and procedures for people, processes and technologies in alignment with business strategies. ISMS should produce a detailed plan about security incidents responses at organization level. This last aspect is important because security risks do not occur just at ICT (Information and Communication Technologies) components level, but also at processes and strategy (and so on) levels. Hence, optimal risk analysis helps managers in their decisions.

Currently, the problems exposed are even complicated by other important issues. Just think about the dynamic nature of security incidents, or inaccurate documentations or assessments of such incidents. Therefore, there is an increasing request of tools and methodologies to enable optimal addressing, understanding and management of cybersecurity risks.

1.1 Managing Security Incidents

Companies information systems are made of different artefacts, security incidents are undesired events that affect one or more of those artefacts. In general, every software, when is built, integrate security controls and should have passed security tests; unluckily is impossible to obtain perfect testing, this means that vulnerabilities can exist in every software product. When those vulnerabilities are discovered, they are exploited by threats to reach and damage the artefacts. In order to minimize the damage, many organizations integrate a Computer Security Incident Response Team (CSIRT) to apply the most appropriate response; furthermore, collaborating with other CSIRT, and thus learning from each other, the response will be much better. But CSIRT are suitable only for large organizations due to its costs, so we need to create a simpler and effective incident management system for small and medium enterprises.

One of the most relevant questions is how to achieve an exhaustive situational awareness to know the status of vulnerabilities, threats and possible incidents. We should be able to apply quick management and responses to incidents, but since the number of those incidents and their interconnection is growing, the problem to face is much more difficult. That is, we should be able to prioritize between hundreds or thousands of incident reports, but this is a task impossible to make manually.

2 GOALS OF THE PROJECT

However, the aim of this project is how to find and select the minimum set of incidents that we must undertake to resolve all existing incidents in a given period of time, taking into account the set of controls that have been affected by them. As can be seen, this is an optimization problem, easy to solve with traditional algorithms when the number of incidents in small, but unsolvable, because of the computational complexity, when that number increases. The need for such a project is justified by the fact that very little effort has been made by the research to join between the resolution of the problem exposed and a performing solution.

In particular, quantum computing is particularly suitable for optimization or machine learning problems. Moreover, information security has been widely impacted by such a computing paradigm, in fact, traditional cryptographic systems are weak against the computing power of a quantum computer; so, there is a need of emergence of a new post-quantum cryptography. In particular, the problem here proposed concerns the optimization of incident response in a risk analysis and management system, where incident response can be optimized by selecting those appropriate controls to perform, being a problem that grows exponentially with the number of incidents.

The solution has been successfully programmed on an IBM quantum computer, using Qiskit as programming technology.

3 METHODOLOGICAL STEPS TO ADDRESS THE GOALS

In this section will be analyzed all the steps followed to address the goal of the project.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3.1 Incident Representation

The first step to face consists in the implementation of a representation of incident themselves. One of the most critical phases for ISMS consists in the incident identification. Once we identify the incident, we have to obtain, categorize and save all the context information, in order to define values for some important parameters.

In this project, every incident is seen as a tuple composed by $< T_{id}, T_{name}, C_{id}, C_{name}, E >$, where T_{id} is the name of the threat that caused the incident, T_{name} is its name, C_{id} is the id of the control affected by the threat, $C_{name}$is the name of the control and, last but not least, E is the estimated resolution time for that specific incident.

The problem we are facing is complicated by the fact that threats are not usually isolated elements, they are often related to each other, so, to define the most efficient way to solve them all in the shortest possible time, we have to consider all these connections (that are common affected controls).

When it comes the need of problem modeling, we have to understand in a proper manner how incidents, threats and controls are related. Each incident involves a single threat, but each threat can affect one or more controls whose implementation must be reviewed. Is very common for different incidents (and eventually different threats) to share the same control. This means that prioritizing the resolution of one of the incidents, we would resolve also all the other incidents that affect the same control.

The exposed problem is an optimization problem, which consists of selecting the minimum set of incidents from among those identified that cover the entire set of affected controls, is easy to solve for small sets of incidents; but become very difficult when we have to work with hundreds or thousands of incidents. Hence, quantum computing is the correct strategy to solve the problem in a performing way.

3.2 Quantum Approximate Optimization Algorithm

The problem to solve will be seen as a combinatorial optimization problem. In those type of problems we want to find an optimal object out of a finite set of objects; in our case the object is a set of incidents and all the possible objects consist in all the subset of incident that we can obtain from the starting incident report.

The algorithm that we will apply is known as Quantum Approximate Optimization Algorithm (QAOA), and it is implemented by Qiskit, the IBM technology that is been used to realize this project.

QAOA uses a unitary $U(\beta, \gamma)$ characterized by the parameters (β, γ) to prepare a quantum state $|\psi(\beta, \gamma)>$. We want the optimal parameter $(\beta_{opt}, \gamma_{opt})$ such that the quantum state $|\psi(\beta_{opt}, \gamma_{opt})>$ encodes the solution to the problem.

The unitary $U(\beta, \gamma)$ has a specific form and is composed of two unitaries $U(\beta) = e^{0-i\beta H_B}$ and $U(\gamma) = e^{0-i\gamma H_P}$ where H_B is the mixing Hamiltonian and H_P is the problem Hamiltonian. Such a choice of unitary derives from a related scheme called quantum annealing.

The state is prepared by applying these unitaries as alternating blocks of the two unitaries applied p times such that: $|\psi(\beta, \gamma)> = U(\beta)U(\gamma)...U(\beta)U(\gamma)|\psi_0>$ where $|\psi_0>$ is a suitable initial state.

3.3 Algorithm Approach

In order to solve the problem, it will be modeled as a *Quadratic Unconstrained Binary Optimization* (QUBO) problem, also known as *unconstrained binary quadratic programming* (UBQP), which will represent all the constraints and constants and can be sent to the QAOA to be solved.

Every problem that follows the QUBO strategy is specified by an Hamiltonian that indicates, trough a summation, the objective and the constraints to be met by the solution. The Hamiltonian is derived as a Binary Quadratic Model (BQM) and converted into a BQM matrix that will be passed to QAOA.

We want to minimize the total time of the issues that are part of the solution. Than we want all controls to have at least one incident related to it that has been selected. If all the controls have been solved, than the issue will be solved as well. In order to build the final QUBO equation we need to add a penalty P (empirically computed adding one at the maximum estimated time registered), which will be used to modulate the weights (bias and coupling weights) of the constraints in the expression. The final equation is:

$$\sum_{i=1}^{N}(x_i * t_i) + \sum_k(\sum_{i,j \in C_k}(-Px_i + 2P * x_i x_j))$$

Where: **i** is an unique identifier of an incident, **k** is a control unique identifier, C_k is a set of incident related to the control with id k, **t** is the estimated time (in hours) to solve an incident, x_i is a binary variable (mapped to a **qubit**) that determines whether the incident i is selected to be addressed or not and **P** is the penalty coefficient.

```
def create_QUBO_problem(linear_terms,quadratic_terms):
    qubo = QuadraticProgram()
    for i in range(1,len(linear_terms)+1):
        qubo.binary_var('x%s' % (i))

    #apply the penalty for each linear term
    for threat_list in actual_controls.values():
        for threat_id in threat_list:
            linear_terms[threat_id-1] -= 73

    qubo.minimize(linear=linear_terms,quadratic=quadratic_terms)

    return qubo
```

Figure 3.3.1: QUBO problem definition

3.4 Quantum Algorithm Execution

Once the matrix has been produced, we just have to send it to the QAOA. After the algorithm has finished, we will get as results all the combination tested and the best one.

```
algorithm_globals.random_seed = 10598

quantum_instance = QuantumInstance(
    BasicAer.get_backend("qasm_simulator"),
    seed_simulator=algorithm_globals.random_seed,
    seed_transpiler=algorithm_globals.random_seed,
)

qaoa_mes = QAOA(quantum_instance=quantum_instance, initial_point=[0.0, 0.0])
```

Figure 3.4.1: Instantiating the QAOA solver

```
qaoa = MinimumEigenOptimizer(qaoa_mes)
```

```
print("Wait.......")
qaoa_result = qaoa.solve(qubo)
print(qaoa_result)
```

Figure 3.4.2: Launching the QAOA

```
Wait.......
optimal function value: -679.0
optimal value: [1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 1.]
status: SUCCESS
```

Figure 3.4.3: The optimal result

```
QUBO variable order: ['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8', 'x9', 'x10', 'x11']
SolutionSample(x=array([1., 1., 1., 1., 1., 1., 0., 0., 1., 0., 1.]), fval=-679.0, probability=0
.08195512500000004, status=<OptimizationResultStatus.SUCCESS: 0>)
SolutionSample(x=array([1., 0., 1., 1., 1., 1., 1., 0., 1., 0., 1.]), fval=-679.0, probability=0.0009765625,
status=<OptimizationResultStatus.SUCCESS: 0>)
SolutionSample(x=array([1., 0., 1., 1., 0., 1., 0., 1., 1., 1., 1.]), fval=-653.0, probability=0.0009765625,
status=<OptimizationResultStatus.SUCCESS: 0>)
```

Figure 3.4.4: Comparison with other solutions

4 PRELIMINARY RESULTS AND FINDINGS

Usually, traditional algorithms that have been proposed for such problem where based on backtracking, dynamic programming or branch and bound strategies. But the problem with this solutions has always been the natural computational complexity of the problem. Quantum combinatorial optimization algorithms, due to their quantum nature and thanks to the concept of superposition, achieve processing similar to multithreaded processing in constant or linear time, depending on the algorithm implemented. For this reasons, it is not brave to assess quantum solutions as the best for the proposed problem.

5 IMPLICATIONS OF THE RESULTS

Quantum computing holds great promise in many areas, such as medical research, artificial intelligence, weather forecasting, etc. But it also poses a significant threat to cybersecurity, requiring a change in how we encrypt our data. Even though quantum computers don't technically have the power to break most of our current forms of encryption yet, we need to stay ahead of the threat and come up with quantum-proof solutions now.

Quantum computers will be able to solve problems that are far too complex for classical computers to figure out. This includes solving the algorithms behind encryption keys that protect our data and the Internet's infrastructure.

It's worth noting that perishable sensitive data is not the main concern when it comes to the quantum encryption threat. The greater risk is the vulnerability of information that needs to retain its secrecy well into the future, such as national security-level data, banking data, privacy act data, etc. Those are the secrets that really need to be protected with quantum-proof encryption now, particularly in the face of bad actors who are stealing it while they wait for a quantum computer that can break the encryption.

There are a lot of questions surrounding quantum computing, and scientists continue to work diligently to answer them. When it comes to the impact of quantum computing on cybersecurity, though, one thing is certain: it will pose a threat to cybersecurity and our current forms of encryption. To mitigate that threat we need to change how we keep our data secure and start doing it now.

6 CONCLUSION

Security management, risk analysis and risk management based on machine learning have become increasingly important. Since the need for proper management of these incidents has grown, there is also much more need of effective end performing processing of the information obtained from incidents themselves. This is true especially by correlating the processing with thousand of incident samples. This need, has come up against current technological limitations, especially those derived from the computational potential required.

The aim of this project has been to design and implement an algorithm based on the new paradigm of quantum programming for performing risk assessment. But there are a lot of other problems to face, for example it would be incredibly useful to integrate machine learning in future releases.