Academy of Science - Independent Researcher

Using a Quantum Genetic Algorithm to Optimize the Spending of a Cyber Security Budget

Author:
Ayush Hariharan

Research Mentor:
Duke Writer

Contents

Abstract

Although computers have increased in computational ability, one category of problems, the NP-Complete problem set, still alludes modern technology. However, quantum computing, a computational method that utilizes quantum mechanics to overcome the limitation of a two-state system, has emerged as a potential solution. The uniqueness of this researcher's current work, Quantum Save, starts by optimizing the Knapsack Problem. Using Qiskit Toolkit and IBM Q32 Simulator, an evolving qubit population was modified using a quantum genetic algorithm. Quantum Save implements a unique disaster algorithm along with tuned hyper parameters for probability amplitude manipulation (a reproduction characteristic). Using this algorithm, knapsack problem optimization increased by 3.75%. Due to Quantum Save's promising results, it was applied to budget allocation for smaller-medium cybersecurity companies to minimize the impact of hacker exploits. With increasing health and financial implications of cyber attacks, technology needs to be enhanced to defend against malware. Using known vulnerabilities and countermeasure effectiveness as Quantum Save optimized the mitigation of a given exploit via the implementation of specific controls while remaining under a budget. By analyzing the distribution of the algorithm's output at each budget range, there was strong evidence which suggested that Quantum Save increased mitigation from attacks and reduced monetary loss faced by a company. On top of that, Quantum Save scales linearly with $\mathcal{O}(n)$ complexity, making it useful in producing recommendation lists with 50-100 countermeasures. Quantum Save's minimal run time and enhanced accuracy can help small-medium companies save millions of dollars by effectively allocating cybersecurity budgets.

Acknowledgements

I would like to thank Duke Writer from the Academy of Science for his generous support throughout the duration of this research endeavor. He has been with me since the beginning of this project and has helped me overcome many obstacles. Throughout this process, he introduced me to the importance of research in computer science and truly conveyed the meaning of inquiry. I would also like to thank Dr. Matt Weber from UC Berkeley for enhancing my understanding of theoretical computer science. He helped refine my project from simply experimental data to theoretical application. He helped me expand the breath of my project so that it could reach a whole new audience.

Background Information

A Technological World

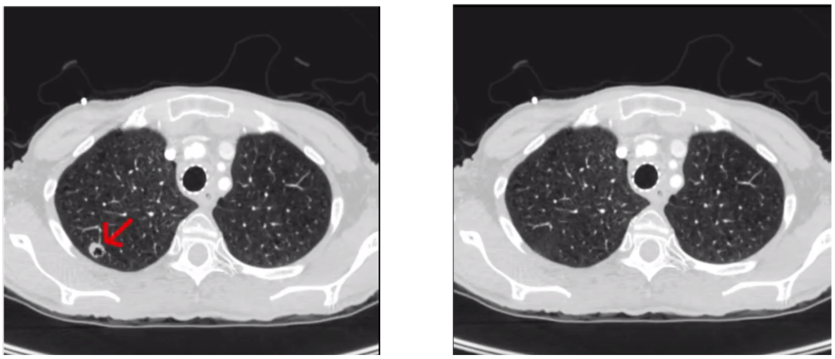
Technology is a powerful agent of change that has shifted the focus of corporations and individuals from face-to-face interactions to online communication. Web applications deployed on accessible backend services such as Heroku or AWS can link family members who live in different parts of the world or companies trying to reach an international consumer base. On top of that, in the public sector, shared hospital networks (usually hosted on cloud services) can help doctors receive real-time electronic health records. The comprehensive streaming of data present in most hospitals can improve their ability to take action regarding critical patient data.

Simply put, technology represents convenience and accessibility in the eyes of individuals, and humans have incorporated this new tool into most aspects of their lives. For example, in the United States alone, Internet traffic for banks grew by 77.6% between July 2000 and July 2001 [10]. On top of that, online retail sales have increased to 269 billion USD in 2004 from a mere 45 billion USD in 2000 [20]. Finally, since the healthcare industry is under constant pressure to reduce costs and improve patient care, information technology can help a hospital manage risks and improve organizational performance [4]. Technology such as computer tomography (CT) scanners and X-ray machines help doctors identify issues much faster. So, in the 21st century, hospitals and businesses alike have seen a growth in users, highlighting the ever growing impact of technology.

The Dangers of Cyberspace

The recent emergence of this online world has been accompanied by increasing attacks on firms and hospitals. However, the reality is that company databases are not secure enough to handle the complexity of the Trojans deployed. In 2015, Chinese hackers stole the background information of 21.5 million federal employee's and 5.6 million fingerprints – credentials that were needed for identity theft. Since the Office of Personnel Management was unaware of this breach, every American who had a background check in the last 15 years was affected [9]. This attack suggests that databases are not strong enough to keep important information secure. Breaches in security happen way too often and the information of 44 million consumers is dangled in the air [17].

The impact of cyber attacks expands far beyond stolen social security numbers and bank account information. The majority of damage occurs when hospital systems are infiltrated in an attempt to confiscate electronic health records – a valuable commodity on the black market. Hacker groups have the potential to disable CT scanners and other technological necessities, placing human lives in serious danger. However, an attacker with access to medical records can do much more than simply sell it on the black market or hold it for ransom. As Israeli researchers Mirsky et al. [11] first proposed, deep learning can be used to add or remove the evidence of medical conditions from volumetric (3D) medical scans [11]. So, to understand the potential danger of existing malware, they infiltrated a local hospital, intercepted CT scans, and altered the images [Fig 1].



(a) Original CT Scan

(b) Altered CT Scan

Figure 1: Modification on CT Scans by Mirsky et al. [11]

By training a generative adversarial network on 888 CT scans, the images were successfully altered with deep learning. The effectiveness of the attack was then verified in a blind and open trial where radiologists were asked to diagnose complete CT scans. In general, the attack had an average success rate of 99.2% for cancer insertion and a 95.8% for cancer removal [11]. This newfound application of malware places thousands of lives in danger and emphasizes the far-reaching repercussions of cyber attacks.

Two Targets

When looking at the targets of these cyber attacks, there are two major firm-types present: bigger targets and small-medium targets. Bigger targets are firms that have a constant influx of money from the government and have substantial changes to their budget each year. Bigger targets can also include cybersecurity departments that receive a large portion of the firm's budget and don't have an immediate limit on monetary spending. On the other hand, small-medium targets, like their name suggests, are smaller firms that are heavily-restricted with the available funding for cyber security, generally working with a fixed budget [6]. They can also include cybersecurity departments that receive a small portion of the firm's budget and face immediate restrictions. This smaller budget forces trade-offs within the company where the chief technical officer has to make important decisions. The bigger targets have layers of security protecting the information making it more secure when compared to a smaller target that does not have the resources to implement that software.

For this reason, approximately 72% of cyber breaches occur at Small-Medium targets making the information more insecure when compared to bigger firms [6].

However, smaller targets do provide substantial benefits to their customers despite their decrease in size. Due to the research and development nature of small-medium targets, they are likely to have more classified data when compared to bigger targets, who are trusted with more public data. On top of that, facilities like hospitals spend so much of their budget on equipment and staffing that little money gets allocated towards infrastructure development. The lack of funding increases the vulnerability of electronic health records. Since the data is more valuable, if there is a breach in a small-medium target, the risks are significantly higher when compared to a breach in a larger company. So, it is important for small-medium targets to implement controls that can mitigate vulnerabilities present in the system. Despite the importance of these controls, most Chief Information Security Officers (CISO) are not confident in patching up vulnerabilities under the budget constraint that is present in the company. For example, in a report published by Deloitte and NAISCO, 75.5% of CISO's cited a lack of sufficient budget as the top challenge (Fielder, 2016). Also, the study finds that 49% of cybersecurity firms only have 6 to 15 specialists working to protect the data.

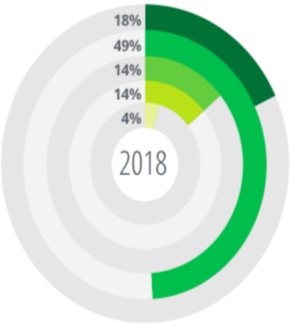
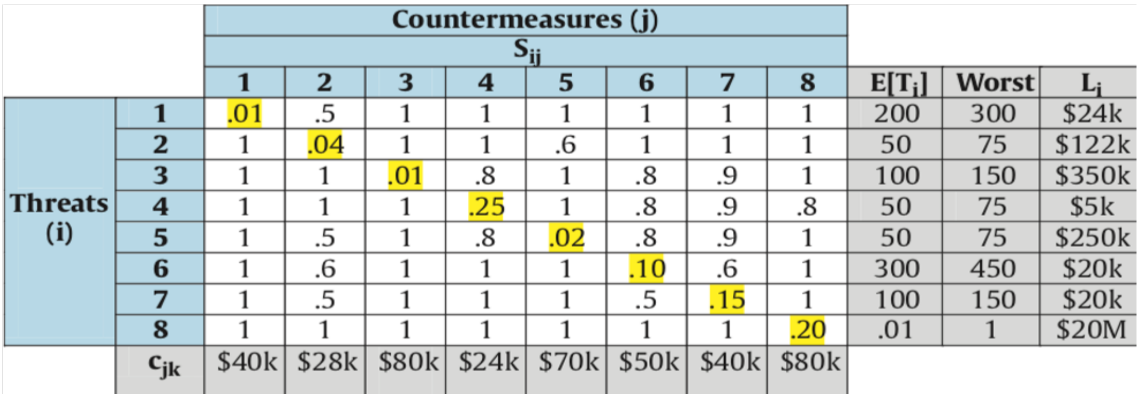


Figure 2: Top Restriction for Small Targets [2]

Each color in the chart [Fig 2] signifies a range of cybersecurity specialists . From the top (18%) to the bottom (4%), the first bar (18%) represents 1-5 full-time employees, the second bar (49%) represents 6-15 full-time employees, the third bar (14%) represents 16-25 full-time employees, the fourth and last bar (4%) represents 26-50 full-time employees. Since there is an abundant shortage of specialists in this field, action needs to be taken. For this reason, it is important to develop a system that will analyze the trade-offs of each control and advise the implementation of the controls that offer the greatest mitigation against an attack.

Endpoint Protection Data

In an attempt to combat this growing issue, third-party organizations, like KnowBe4, publish statistics on the specifics of each attack in Endpoint Protection Reports. The Endpoint Protection Report provides information regarding the frequency of cyber attacks in a given year and the effectiveness of potential countermeasures which companies might implement to protect themselves. Using the 2009 endpoint protection report on security breaches and cost of downtime, Rakes et al. [18] from the Pamplin College of Business developed a dataset which consolidated the information in the report [18]. Qualitative attributes present in the endpoint protection report, such as countermeasure effectiveness, were expanded into quantitative values such as survival probabilities. The dataset expresses the relationships between a variety of different threats and countermeasures.



		Countermeasures (j)										
		S_{ij}										
		1	2	3	4	5	6	7	8	$E[T_i]$	Worst	L_i
Threats (i)	1	.01	.5	1	1	1	1	1	1	200	300	$24k
	2	1	.04	1	1	.6	1	1	1	50	75	$122k
	3	1	1	.01	.8	1	.8	.9	1	100	150	$350k
	4	1	1	1	.25	1	.8	.9	.8	50	75	$5k
	5	1	.5	1	.8	.02	.8	.9	1	50	75	$250k
	6	1	.6	1	1	1	.10	.6	1	300	450	$20k
	7	1	.5	1	1	1	.5	.15	1	100	150	$20k
	8	1	1	1	1	1	1	1	.20	.01	1	$20M
	c_{jk}	$40k	$28k	$80k	$24k	$70k	$50k	$40k	$80k			

Figure 3: Compiled Cybersecurity Data by Rakes et al. [18]

When analyzing the data set [Fig 3] compiled by Rakes et al. [18], there are three major aspects that are taken into consideration: the survival probabilities, the general attack statistics, and the cost of implementing the countermeasures. The survival probabilities, which is the array in white, represent the probability, $P(X, Y)$, of an attack (X) succeeding when a certain countermeasure (Y) is implemented. Although all countermeasures are most effective against a specific threat, the array has off-diagonal terms because some countermeasures work not only against their primary threat, but because they provide additional protection against other threats as well. The general attack statistics, which are the three gray-shaded columns, represents the expected number of attacks in a given year ($E[T_i]$), the worst-case number of attacks in a given year ($W[T_i]$), and the damage incurred from each successful attack (L_i). Finally, the cost of implementing the countermeasures, the gray-shaded row, represents the monetary cost a company would have to pay in order to install that specific countermeasure.

Linear Optimization Model

Using this compiled data set, Rakes et al. [18] developed a linear optimization model that attempted to allocate budgets for companies of all sizes. The model selected countermeasures based upon quantitative factors such as threat levels and countermeasure effectiveness [18]. The model took in the budget of a company as the input and attempted to choose appropriate countermeasures to protect that firm. Depending on the countermeasures chosen, the model would output the expected damage to the company in the case of an attack. In order to simulate the decision making process of firms, Rakes et al. followed three major steps. The first involved determining the probability P_i that a threat i would cause monetary loss.

$$P_i = \prod(1 - Y_i e_{ij}) \tag{1}$$

However, in order to linearize the model, the effectiveness, e_{ij}, was replaced with the proportion of threats that will survive an attack. On top of that, constraints were added to ensure that the proportion of threats which survived the first countermeasure would be assigned to the second countermeasure (Rakes, 2010). The process repeated for all countermeasures in which the binary selector, Y_i, was true. After the assignment process, the model estimated the damage to a company by surviving threats. The monetary loss (both explicit and implicit) was calculated by multiplying the number of surviving attacks by the loss per successful attack. This allowed Rakes et al. to approximate the amount of money saved by implementing each countermeasure. Once calculating the monetary loss, the model ensured that the cost of implementing the countermeasures would remain under a specified budget.

Using the compiled data set [Fig 3], Rakes et al. [18] tested their linear optimization model on both the expected-case attacks and the worst-case attacks. The researchers looked at the relationship between company budget and monetary loss in the case of an attack. The loss was used as an indirect measure of the algorithm's decision making capabilities. The monetary loss was predicted at each of 16 different budgets, comprising bigger and small-medium targets [18].

When analyzing both the expected-case relationships and worst-case relationships of the linear model [Fig. 4], the budget and monetary loss seem to have an inverse relationship. With a higher budget, the algorithm can implement more countermeasures and the monetary loss faced by the companies is reduced. However, with small-medium targets, the linear optimization model predicts a high monetary loss in the case of an attack. So, an algorithm is needed that can help small-medium targets effectively allocate their cybersecurity budgets since it is these smaller targets that are more susceptible to malware.

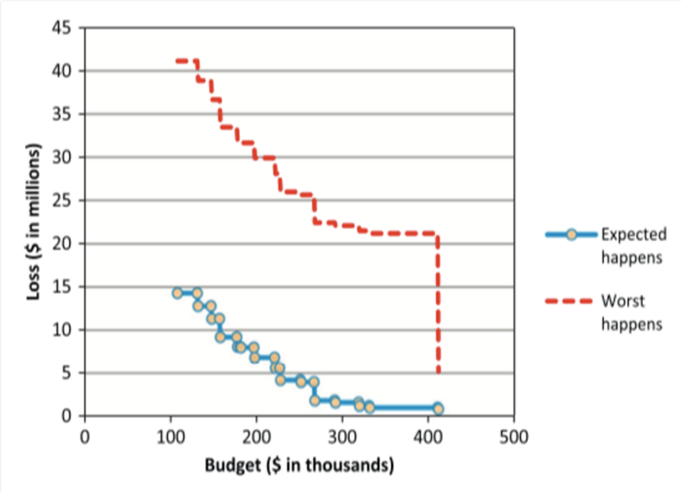


Figure 4: Model Allocation at Each Budget Range [18]

Knapsack Problem

To improve the budget allocation of small-medium targets, the budget allocation problem has to be analyzed from a different angle. In fact, the complex trade-offs caused by a budget constraint can be simplified to a single-choice Knapsack problem. The knapsack problem is a NP-Complete optimization problem with the goal of finding, in a set of items with given values and weights, the subset of items with the highest total value, under a weight restriction [13].

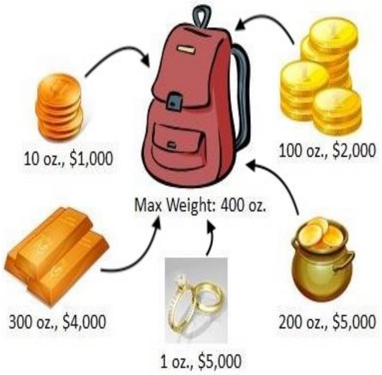


Figure 5: Knapsack Problem [16]

In this problem, a value has to be optimized while remaining under a certain constraint. A common interpretation of the knapsack problem is to design a system that can handle a large influx of data while remaining under corporate budget. Other interpretations include determining which parts to purchase in order to increase the fuel efficiency of a car. In the case of cyber security budget allocation, the mitigation of an attack has to be maximized while remaining under the budget constraint provided to the cybersecurity department of the small-medium target. Computer scientists have successfully solved this Knapsack problem classically; however, the algorithms are generally slow and time-consuming. With quantum computers providing a drastic increase in computational ability, new approaches can be designed to take advantage of a quantum computer's unique properties like superposition and entanglement.

Quantum Systems

With the complexity of mathematical problems, quantum computing can be a solution to avoid long periods of computation. A quantum computer uses fundamental principles of quantum mechanics, including superposition and entanglement, to overcome the limitation of a two-state classical system. In a traditional computer, data is encoded in bits, 0 and 1, which represent the on and off positions of a transistor; however, a quantum computer uses quantum bits, or qubits, to represent a weighted probability of the two initial states. Each qubit is mathematically notated as a vector, with a horizontal component that represents the probability of being a 0 and a vertical component that represents the probability of being a 1. The horizontal component will be referred to as α, and the vertical component will be referred to as β. Although it is impossible to determine the exact state of a qubit, due to uncertainty and superposition, vector notation allows scientists to approximate this value using the direction and magnitude of the qubit. In a quantum system, or a space with multiple qubits, a major characteristic is that the magnitude of all of the probability vectors in the system must equal 1.

Another important aspect of any quantum system is the idea of measurement, which allows for a transformation of quantum states into classical states. A quantum entity, or system, only comes into existence and acquires definite properties when an observation or measurement is made [7]. An important part of quantum mechanics are wave functions which give the mathematical probabilities of the state of a quantum entity before an observation has made [7]. Each possible probability in the wave function is commonly referred to as an Eigenfunction.

In order to obtain a measurable value from the system, the wave function must be collapsed into a specific Eigen function, as can be seen in Figure 6. In quantum mechanics, the possibilities of a wave function are localized into a single particle by adding energy and this localization results in measurement. However, the results from a measurement of the wave function is not random; every Eigen function has a certain probability of occurring. For quantum computing, there is a certain probability of returning 0 and a certain probability of returning 1.

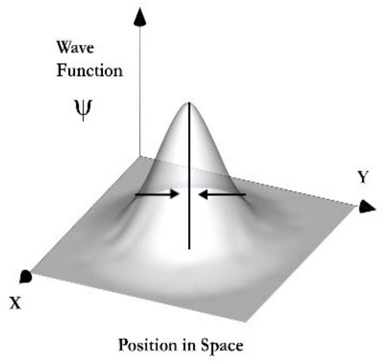


Figure 6: Quantum Measurement [1]

Quantum Genetic Algorithms

A possible application of quantum computing involves a quantum-genetic algorithm. A standard genetic algorithm is a population-based method which consists of 5 crucial components: initialization of a population, evaluation, parent selection mechanism, variation operators, and survivor selection [8].

Algorithm 1 Standard Genetic Algorithm

```
$t \leftarrow 0$
$f(x) \leftarrow$ Evaluation function
$population \leftarrow$ Initial Population
**while** $generation <$ final generation **do**
    $values \leftarrow f(population)$
    $survivors \leftarrow \max(values)$
    **for** $individual \in survivors$ **do**
        Mutate the Survivors
        $population \leftarrow$ Survivor Offsprings
    $generation+ = 1$
```

Throughout the course of Algorithm 1, the population should "evolve" towards an optimal solution. Although genetic algorithms are a good way to optimize problems, classical computers require additional steps in order to mutate the population. Genetic operators such as crossover probability and mutation effects have to be explicitly defined for the population to mutate. On top of that, these genetic operators require hyper parameter turning and will not accurately represent the random variability in most populations.

When applied to quantum mechanics, however, previous research has indicated that a quantum genetic algorithm can be used based upon the concepts of quantum bits and quantum superposition states. Using qubit encoding, where each qubit uses a vector to represent binary code, a quantum algorithm can use a quantum system with multiple qubit vectors as an initial population [21]. Specifically, each quantum gene is represented by a qubit vector, each quantum chromosome is represented by a bra vector, and each quantum population is represented by a column of chromosomes. When the quantum population is first created, each vector is initialized with a linear superposition of states, essentially setting both α and β to $\sqrt{2}/2$, so that the probability of getting a 0 or a 1 during measurement is the same.

Once the initialization of the population has been completed, the entire quantum population is measured. Measurement involves the evaluation of each individual in the population, as explained in Algorithm 1. However, the evaluation function will differ tremendously depending on the application of the quantum genetic algorithm. The qubit string which produces the highest profit while remaining under the constricting value is stored as the best individual in that generation. The fitness of the best population is then plotted on a curve for future use. In order to create a new generation from the existing quantum population, each qubit's probability amplitudes are manipulated by a constant (a hyper-parameter in the algorithm). When a new generation is created, the inherent randomness of quantum computing is utilized since this randomness creates natural variability in the population. The population will eventually reach a local maximum and the evolutionary algorithm is terminated [15]. Thus, the theory of the genetic algorithm remains the same but a QIGA manipulates qubit vectors, instead of classical values, to create increase optimization. Higher-Order quantum genetic algorithms use quantum registers to handle the encoding instead of isolated qubit vectors [15].

Disaster Algorithm

The current setup of the quantum genetic algorithm does not account for the stabilization that occurs when an environment reaches a local optimum. A local optimum is when the genetic algorithm prematurely converges onto a single value for a certain period of time giving the illusion that the absolute maxima, or extremity point has been determined [15]. So, one caveat of the genetic algorithm is the time of convergence in later states of the algorithm.

Fu et al. [21] addressed this issue and improved local search performance by implementing a self-adaptive rotating angle strategy and a disaster algorithm [21]. The rotating strategy exploited different concepts of linear algebra to decrease the number of parents with each call of the algorithm. The disaster condition, however, imposed a self-made disaster once the genetic algorithm reached a premature convergence value. For example, in some functions, there are various relative maxima and the genetic algorithm might mistake a relative maximum for an absolute maximum. In a quantum system, the disaster was usually resetting the probability amplitudes of some vectors to a linear superposition of states. This prevents premature convergence and ensures that the highest value for the knapsack function is returned.

Although Fu et al. [21] did not test their algorithm using the knapsack problem, their method required similar computational abilities. After testing, researchers concluded that the addition of the rotating technique and the disaster condition improved the times of convergence of the quantum-inspired genetic algorithm, and returned a higher optimization value [21]. However, the research that was performed by this team used isolated qubit vectors to encode qubit strings, as common with QIGA algorithms, and the simplicity of the design could have caused a lower optimization value than initially expected.

Uniqueness of This Researcher's Current Work

The new research performed addresses the flaws in previous research and allows for greater theoretical optimization of the knapsack problem. In this new algorithm, Quantum Save, the higher-order quantum genetic algorithm is implemented on the IBM Q32 simulator which allows the algorithm to better simulate a quantum state. Quantum Save uses the increased complexity of the Q32 simulator to better mimic the random variations that occur in a genetic algorithm. In addition to that, although Kucharski and Nowotniak [15] provided a range for the amount of probability amplitude manipulation, they didn't provide a certain value. So, Quantum Save will consist of tuned hyper parameters for which the optimization of the Knapsack problem is the greatest. Also, with Quantum Save, a disaster condition will be added to the higher-order QGA to prevent premature convergence. In fact, social disasters technique, which applies a catastrophic operator once a local maximum is reached is one of the best methods to obtain the global extremity point instead of the local maxima [19]. In Quantum save, the disaster condition will implement a simple quantum catastrophe operator. Therefore, by tuning the higher-ordered QIGA, the probability of reaching a global maximum will increase drastically, and the optimization value for the knapsack problem will be much higher.

In addition to that, Quantum Save can help hospitals and defense contractors save millions by allocating the budgets of small and medium sized targets. In previous research, the cyber security investment model was based upon simulated values and game theory. And with the existing optimization models, like the one proposed by Rakes et al. [18], the model performs significantly better for bigger targets since there is more budget to work with. The model does not face the full effect of the budget constraint and can easily choose countermeasures to maximize mitigation. However, it is the smaller-medium sized targets who are the most vulnerable to malware from adversaries. So, Quantum Save uses threat factors and mitigation values from real-life case studies to improve the budget allocation of these small-medium sized targets. Specifically, the data set compiled by Rakes et al. [18] in Figure 3 is inputted into the quantum algorithm. Researching this topic will allow the scientific and engineering community to accomplish countless tasks that have a limitation factor and an optimization value. Quantum Save can also improve the cybersecurity defenses of hospitals and defense contractors.

Experimental Design

Variables

In this research problem, the extent to which Quantum Save can optimize the spending of a cybersecurity budget is investigated. Using the data set defined by Rakes et al. [18], Quantum Save will be implemented and an optimal investment strategy can be determined. The project will consist of two phases where the first phase involves tuning the Quantum Save algorithm with the optimal probability amplitude manipulation and implementation of the disaster algorithm. In this phase, it was hypothesized that the probability amplitude manipulation along with the disaster algorithm implemented on a POWER 9 architecture would result in the optimization of the Knapsack problem. For the first phase, the independent variable is the manipulation of the probability amplitudes and the implementation of the disaster algorithm while the dependent variable is the optimization of the Knapsack problem.

The second phase of the project involves applying the Quantum Save algorithm to the cyber security modeled previously defined and see if this algorithm can provide an optimal investment strategy for CISO's. In this phase, it was hypothesized that if the Quantum Save algorithm is implemented in the cyber security investment problem, then the investment strategy that is obtained will be better when compared to previous case studies (specifically, the linear optimization model by Rakes et al. [18]). This is because Quantum Save has a better optimization of the knapsack problem when compared to that of previous research. Also, quantum computing would allow for a better simulation of random variability in the genetic algorithm. In this phase, the independent variable is the implementation of the QIGA algorithm with a cyber security purpose while the dependent variable is the investment strategy that is obtained after implementing the QIGA algorithm.

Quantum Save consists of six major steps and these individual steps are performed multiple times throughout the evolution process. Every time these steps are performed is called a generation and through multiple generations, the population should evolve to produce a greater optimization of the knapsack function. The steps in each generation derive from Darwin's theory of evolution which states that evolution is a cause of overpopulation, competition between individuals of the species, and eventual survival of the fittest. However, before implementing the QIGA-2 algorithm on the IBM Q32 computer, various libraries including Qiskit by IBM and NumPy must be imported in order to connect with the quantum computer.

Phase 1 of Quantum Save

The first section of the quantum genetic algorithm is reliant on the idea of over- population and natural variability in genes. As Darwin concluded, evolution is spurred when there are too few resources for too many individuals and random variability in the population cause some individuals to be better than others. In the first step, the quantum population is created and initialized. In this case, the population is an array of 32 qubits, the maximum number of qubits available on the IBM Q32, and each quantum chromosome represents a row in that array.

Algorithm 2 Quantum Save – Phase 1

```
$n \leftarrow$ number of countermeasures
$b \leftarrow$ budget
$probabilityAmplitude \leftarrow [0.5, 0.5, 0.5, 0.5]$
$generation \leftarrow 0$
**while** $generation < 10$ **do**
    $quantumRegister \leftarrow 0$
    **while** $quantumRegister <$ n **do**
        $classicalRegister \leftarrow$ classicalRegister[n]
        $quantumCircuit \leftarrow$ classicalRegister + quantumRegister
        $quantumCircuit \leftarrow$ probabilityAmplitude
        $job \leftarrow measure(quantumCircuit)$
    $quantumRegister{+}{+}$
```

The array of 32 qubits is arranged to simulate a Think-Tank of four Chief Security officers. The goal of the Think Tank is to create a recommendation list of 8 countermeasures which defense contractors and small-medium targets across the world can implement in order to protect themselves from foreign attacks. Each of the four rows in the qubit array, or matrix, represents a Chief Security Officer in the think-tank, and each of the eight columns in the qubit array represents a countermeasure on the recommendation list. If the qubit returns a value of 1, then the countermeasure is added to the recommendation list. On the other hand, if the qubit returns a value of 0, then the countermeasure is removed from the recommendation list (Fig. 7).

After the quantum population is created, each qubit register in that population is initialized with a linear superposition of states. By starting out with an equal chance of getting one and zero, there is natural variability in the population and the genetic algorithm can be more effective. In the case of the cyber security application, the linear superposition of states that the quantum population is initialized with represents an equal probability of a defender implementing a certain control and not implementing that control. Therefore, there is natural variability in the population and the genetic algorithm is effective.

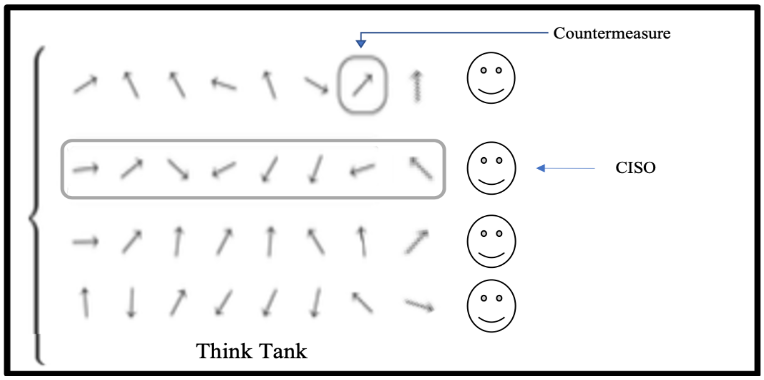


Figure 7: Quantum Population on Q32

Phase 2 of Quantum Save

Now that the population has been created and has been initialized with a linear superposition, the second section of the quantum genetic algorithm can begin. This section the individuals with the best random attributes are chosen and the fittest individuals in the population, or those who pack the knapsack the best, will remain for the next generation. After the initialization of the quantum population, it will be classically measured by collapsing the existing wave function. Depending on the alpha and beta values present in each qubit register, a binary string of size two, either 00, 01, 10 or 11, will be obtained from each register

Algorithm 3 Quantum Save – Phase 2

```
$CISO \leftarrow 0$
while $CISO < 4$ do
    $binaryString \leftarrow$ ""
    $result \leftarrow$ ""
    while $result$ in job do
        $binaryString \leftarrow binaryString +$ Most Frequent String
    $countermeasure \leftarrow 0$
    $knapsack \leftarrow 0$
    while $countermeasure$ in binaryString do
        $knapsack \leftarrow knapsack + countermeasure$
        if $knapsack > b$ then
            $knapsack \leftarrow knapsack - countermeasure$
        $profit \leftarrow$ amount saved[$countermeasure$]
    $bestIndividual \leftarrow max(profit)$
```

During each fitness evaluation, or the conversion between quantum and classical states, the qubit register is measured 1024 times and the most consistent binary string is returned. The binary strings from every four registers will then be combined until four classical strings are obtained, one for each individual in the Think Tank.

After quantum measurement, using the Copenhagen interpretation, is complete, the binary strings are evaluated in order to determine the best individual. Depending on the binary string, which essentially represents the controls that are implemented by the defender, a certain mitigation will be received. In this case, if there is a 1 that means that the defender chose to implement a specific control. On the other hand, if there is a zero, then the defender didn't implement a certain control. Apart from the profit value, a total weight for the knapsack was also obtained, or the total cost of implementing all of the controls, and that total cost was compared to the cost cap, or the budget, set before the first generation. After each binary string is evaluated, the binary string, and respective individual, with the highest profit value is stored as the best chromosome. With the application to a cyber security investment problem, the best chromosome will be the defender that implemented the controls that resulted in the greatest mitigation of the attack.

Phase 3 of Quantum Save

Since the best strategy to mitigate an attack on the defender's small-medium enterprise has been determined, the probability amplitudes of each qubit register will be manipulated in the following manner.

Algorithm 4 Quantum Save – Phase 3

```
$binaryArray \leftarrow binaryString.split()$
$CISO \leftarrow 0$
while $CISO < 4$ do
    $countermeasure \leftarrow 0$
    while $countermeasure$ in $binaryArray$ do
        if countermeasure != bestIndividual then
            $probabilityAmplitude \leftarrow$ manipulate[$probabilityAmplitude$]
        $probabilityAmplitude \leftarrow probabilityAmplitude$
=0
```

Initially, the first binary pair in the best individual will be obtained and depending on the value of that pair, the probability distribution will be normalized to one. On the other hand, all of the other probability amplitudes will decrease by a factor of *probabilityAmplitude*, a hyper parameter in the algorithm. In this case, the probability of a defender implementing the successful controls increases while the probability of a defender implementing the unsuccessful controls decreases. The factor of probability manipulation, *probabilityAmplitude*, is a hyper parameter of this algorithm and this hyper parameter was tuned by testing out different values for and observing the optimization of the knapsack problem. In the end, the controls that resulted in higher mitigation of an attack were encouraged while controls that resulted in lower mitigation of an attack were discouraged (Fig 8).

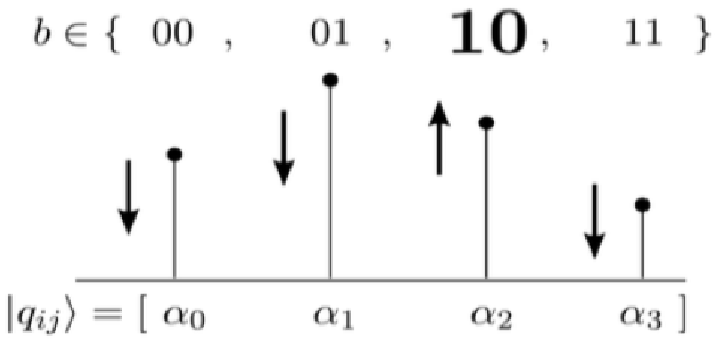

$b \in \{ 00, 01, \mathbf{10}, 11 \}$
$|q_{ij}\rangle = [\alpha_0 \quad \alpha_1 \quad \alpha_2 \quad \alpha_3]$

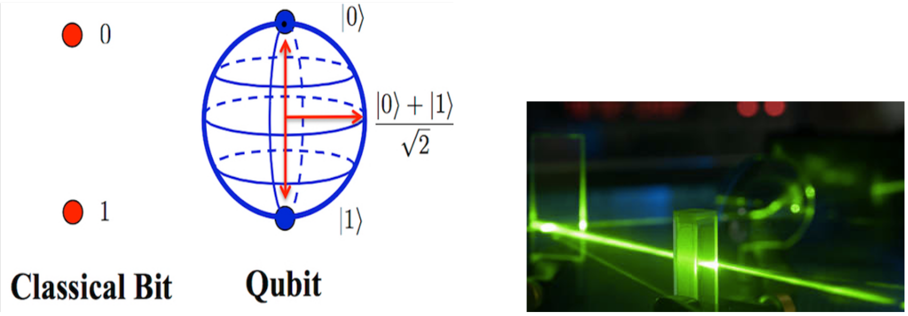
Figure 8: Manipulation of Probability Amplitudes

At this point, the algorithm has completed one generation of fitness evaluations. In order to obtain an accurate optimization of the knapsack problem, this process needs to be repeated for 10 generations in order to get a higher optimization of the knapsack problem. At the end of these ten generations, the data of final profit outputted by Quantum Save will be obtained and that data can be compared to that of previous research. However, now that the probability amplitude manipulation is complete and the optimal value has been determined, a disaster algorithm will be implemented. In this case, the disaster algorithm will simply reset the population to a linear superposition of states in the middle of Quantum Save. However, the generation that the disaster algorithm is implemented, is a hyper parameter of this algorithm and this hyper parameter was tuned by testing out different generations. In the end, the location of the disaster algorithm was chosen by the overall mitigation of an attack by the defender.

After one instance of the algorithm, or 10 generations, has completed its run-time, the data of final profit outputted by Quantum Save will be obtained and that data will be compared to that of previous research. On top of that, the different hyper parameters including the value and the generation for the disaster algorithm will be tuned.

Why Quantum Computing

With this quantum algorithm, one important aspect is the evolution of the population. In order for a population to evolve, there needs to be random variation in the population so that certain traits emerge over others. If every generation contained the same traits, then the entire population would either die or survive. Instead, mutations and gene crossovers drive the idea behind natural selection. With this evolution process, a quantum computer would significantly enhance the effect of a genetic algorithm.



(a) General Qubit Properties

(b) IBM Quantum Simulator

Figure 9: Power of the Qubit

On a traditional classical computer, a bit is isolated to either 0 or 1, but a qubit uses quantum mechanical properties to simulate a weighted probability distribution between 0 and 1. Each qubit is initialized with a 50% chance of returning 1 and a 50% chance of returning 0. As the algorithm progresses, the probabilities change, with a potential 90% chance of returning 1 and a 10% chance of returning 0. In a classical genetic algorithm, random variation is controlled by unique constants. However, by using quantum computing, the random variation is inherent within the computational method. The measurement value of a qubit is never concrete but fluctuates based upon probability amplitudes. More specifically, the IBM quantum simulator uses a dilution refrigerator to suspend an electron and shoot a pulse at the suspended electron. If the electron is spin-up, the qubit returns a zero. On the other hand, if the electron is spin-down the qubit returns a one. This allows the IBM Q32 simulator to accurately simulate a quantum state.

Results

Tuning Hyperparameters

The first phase of data collection involves tuning the actual hyper parameters of the quantum genetic algorithm. One hyper parameter that needs to be tuned is the µ value that describes how much the probability amplitudes are manipulated when the quantum population is evolving. This hyper parameter is important because it is fundamental when describing the probability amplitude manipulation that occurs in each generation of the genetic algorithm. It helps determine which cyber security controls are optimal and successful in mitigating attacks. Depending on the value for µ, the optimization of the knapsack problem can either increase or decrease. In this research, four different values for µ were tested including 0.8, 0.9, 0.95, and 0.99 and the final optimization of the knapsack problem was recorded.

Table 1: Results of Quantum Save with Varying Hyperparameters

Statistics	**Probability Manipulation**					**Disaster Algorithm**			
Trial	80%	90%	95%	99%	*Baseline*	5	6	7	*Baseline*
1	8957	9528	9416	7004	5662	7503	7689	8853	9528
2	7509	7723	8039	5885	6406	7017	7117	8432	7723
3	8631	10436	7108	7261	6920	8642	7277	9290	10436
4	8005	9352	7040	6135	5980	9920	7793	9696	9352
5	8149	7566	7462	8179	9678	8390	8398	8722	7566
6	8299	7324	6941	7030	8775	9323	8031	8154	7324
7	7798	8368	9753	7076	9258	8762	8338	8459	8368
8	9303	8504	7077	8418	7070	7945	8484	6486	8504
9	7799	6020	7161	8649	6954	7044	8232	7910	6020
10	8230	7008	9684	9064	4097	8448	8008	7736	7008
AVG	8150	8252	7975	7641	7086	8518	8235	8292	8252
SE	156.51	310.67	266.80	293.33	414.21	236.19	241.81	214.09	310.67

*** The first 10 trials out of 15 total are shown. All values are in dollars. The baseline for the Disaster Algorithm is Quantum Save's performance with simply probability amplitude manipulation. The baseline for the Probability Manipulation is Quantum Save's performance without the manipulation.

These specific values were chosen for a reason when dealing with probability amplitude manipulation. At first, based upon patterns observed in Nowotniak and Kucharski's initial research [15], higher probability manipulation values, such as 99% would be around the optimal manipulation value since that would drastically increase the probability of implementing a successful control. However, after running the algorithm under this high manipulation value, it was realized that an excessively high value would result in over fitting, where each defender was too reliant on the patterns of the best individual. So, the value was reduced to 80%. After running the algorithm again, it was determined that the defenders in this generation are not as reliant on the patterns discovered in previous generations and a lower optimization was occurring. So, as a compromise, the value was raised to 95% and then lowered back down to 90% to get a variety of probability amplitude coefficients.

After this data was collected regarding the probability amplitude manipulation, another vital hyper parameter was the generation at which to implement the disaster algorithm. Depending on the generation that the disaster algorithm was implemented at, the optimization of the Knapsack problem and the choosing of countermeasures changed. For example, if the disaster algorithm was implemented towards the later generations, then the behaviors of the defenders would not impact the overall optimization of the Knapsack problem and the company would get stuck at the local maximum. On the other hand, if the disaster algorithm was implemented towards the earlier generations,t hen the company would not receive the higher level of mitigation that the previous generation had gotten. Either way, there is a trade-off between getting stuck at a local maxima and not even reaching that local maxima. To resolve this trade-off, it is important to tune the disaster algorithm hyper parameter as done in Table 1. The data collection started at a disaster step of 7 and then slowly decreased to 6 and eventually to 5. For each of these disaster steps, Quantum Save was run, with all 10 generations, and the final output was recorded.

Cybersecurity Application

Now that the hyper parameters have been tuned, Quantum Save can be implemented in the context of a cyber security investment problem where a small-medium target has to implement controls that result in the highest mitigation. The total mitigation across 10 generations was recorded and Quantum Save was run 10 times in order to determine whether the mitigation is better than that of existing case studies, specifically the linear optimization model proposed by Rakes et al. [18].

Table 2: Quantum Save vs. Other Budget Allocation Models

Budget	Quantum Save	Linear Algorithm	Brute Force Algorithm
80k	26.57 M	32.20 M	14.29 M
108k	14.26 M	14.29 M	11.24 M
132k	11.95 M	12.78 M	8.294 M
148k	11.31 M	11.32 M	5.244 M
158k	8.340 M	9.184 M	5.244 M
178k	7.714 M	8.071 M	5.244 M

The data shows the relationship between the budget of the company and the monetary loss due to a cyber attack. The relationship between the two variables and the effectiveness of Quantum Save will be investigated later. Now that the data concerning Quantum Save's accuracy has been collected, the next step of the research centered around the run time of the algorithm. When analyzing an algorithm's effectiveness, the speed of the algorithm needs to be taken into account. More specifically, data was gathered surrounding Quantum Save's scalability, or how the size of the input (number of countermeasures in the recommendation list) affected the run time of the algorithm. The data collection process involved obtaining the run time of the algorithm across 10 trials for each permutation of input sizes. The run time was measured using the internal clock on both the quantum simulator and the classical machine. The input sizes varied anywhere from 1 countermeasure in the recommendation list to 8 countermeasures in the recommendation list. After collecting data about the scalability of Quantum Save, the same procedure was completed for the Brute Force algorithm for comparison purposes.

Analysis

Hyperparameters

Throughout this research, there were two major changes that distinguished Quantum Save when compared to existing algorithms. The first is the utilization of quantum computing, or the IBM Q32 quantum simulator. To evaluate the effectiveness of quantum computing in optimizing the Knapsack problem, or more accurately the cybersecurity defense problem, the evolution of Quantum Save was compared to that of Kucharski and Nowotniak [15]. Kucharski and Nowotniak [15] used a classical implementation of the quantum-inspired genetic algorithm to find a solution to the knapsack problem.In order to establish the significance of Quantum Save in the context of the Knapsack problem, the baseline optimization that was obtained, without any probability amplitude manipulation, was compared to the highest value obtained by Nowotniak and Kucharski [15], $5709.

In order to compare the results of Quantum Save when compared to previous research, a 1-sample t-test with a significance level of 0.05 was run on the baseline probability manipulation (Table 1) and the results of the test were used to determine whether Quantum Save on the IBM Q32 simulator was significantly better when compared to Kucharski and Nowotniak [15]. Since both of the p-values (one-tailed and two-tailed) obtained from the 1-sample t-test, *0.003* and *0.005* respectively, were less than the established significance level of 0.05, the null hypothesis could be confidently rejected. If the null hypothesis was true, as the proposed algorithm didn't perform any better than that of previous research, there would only be a 0.274 or 0.548 percent chance of obtaining a profit value this high or higher based upon random sampling variability alone. Therefore, there is enough statistical evidence to suggest that Quantum Save on the IBM Q32 computer performed significantly better than that of previous research. In the end, Quantum Save, on the IBM quantum simulators, likely performed better due to the simulation of more qubits and the existence of four quantum chromosomes in the population as opposed to three. These factors allowed for more genetic variability and increased the chance of finding the non-obvious solutions to the knapsack problem. Different routes were taken and ultimately, a higher optimization of the knapsack function was achieved.

However, Quantum Save was improved by tuning hyper parameter values including the value of µ which described the amount of probability amplitude manipulation and the generation for which to implement the disaster algorithm. First, when regarding the manipulation of probability amplitudes, an analysis of variance test, or an ANOVA test, was run to determine whether the change in probability amplitudes did in fact achieve a statistically significant change in the optimization of the knapsack problem. The significance level of the ANOVA test was 0.10 and that represented the alpha power of this statistical test. When implementing the probability amplitude manipulation, the goal was to simply increase the optimization of the knapsack problem. So, the alpha power of the test did not need to be extremely high so a value of 0.1 was chosen. Since the p-value of *.052* is less than the significance level of 0.1, the null hypothesis could be confidently rejected. If the null hypothesis was true, as the proposed algorithm with manipulation of amplitudes did not perform any better than the algorithm without the manipulation of amplitudes, there would only be a 5.27% chance of obtaining a profit value this high or higher based upon random sampling variability alone. Therefore, there is enough statistical evidence, within the power of this ANOVA test, to suggest that the manipulation of probability amplitudes increased the optimization of the knapsack problem. Since the ANOVA test does not provide the greatest value for µ, that had to be calculated manually. When looking at the data table, it could be observed that a µ value of 0.9 or a 90% manipulation of amplitudes resulted in the highest optimization of the knapsack problem. In order to confirm the values obtained from the ANOVA test, the maximum profit over the probability amplitudes were observed.

The second tuned hyper parameter was the generation on which to implement the disaster algorithm. To determine whether the disaster algorithm achieved a statistically significant change in the optimization of the Knapsack problem, another analysis of variance test, or an ANOVA test, was run. Again, like with the manipulation of probability amplitudes, the the significance level of the ANOVA test was 0.10 and that represented the alpha power of this statistical test. With a p-value of *0.848,* it appears that the disaster algorithm does not significantly increase the optimization of the Knapsack problem. However, in Table 1, a disaster step of 6 does increase the optimization of the knapsack problem by around 3.22%. Although this is not statistically significant, there is some improvement. The disaster algorithm will help only when Quantum Save reaches a local maxima, which might not occur in every trial. So, with any improvement, the disaster algorithm is considered successful. Also, when looking at Table 1, the standard error with the disaster algorithm is much lower when compared to the baseline. This suggests that the disaster algorithm is preventing Quantum Save from getting caught on the lower local maxima. Now, the probability of finding that non-obvious solution increases drastically.

Accuracy of Quantum Save

The second uniqueness of Quantum Save when compared to previous research, such as the linear optimization model [18] is the implementation of the QGA algorithm in cybersecurity budget allocation. However, when analyzing the effectiveness of Quantum Save in the realm of cybersecurity budget allocation, there are two distinct features that need to be considered: the accuracy and the run time. With the accuracy, the budget allocation techniques of Quantum Save can be compared with the techniques from linear optimization models [18]. The techniques will be quantified by analyzing the amount of money that a small-medium target will lose when implementing the countermeasures suggesting by each respective algorithm. The better algorithm will be the one that results in lower monetary loss in the case of an attack by foreign or domestic adversaries. From a simple eye-test on the data presented in Table 2, it appeared that my algorithm performed better when compared to previous research [18] at every budget range. However, the extent of that improvement varied significantly (Table 3).

Table 3: Percent Improvement by Quantum Save

Budget	% Improvement
80k	17.48
108k	0.24
132k	6.53
148k	0.05
158k	9.19
178k	4.42

When looking at Table 3, it was determined that Quantum Save performed better when compared to the generalized linear optimization model. But, the extent of that improvement varied from 0.05% to 17.48%. Since the discrepancy in the percent improvement is so high, another statistical method had to be taken to ensure that the improvement was not caused by random variability. The distribution of Quantum Save's outputs at each budget range was analyzed using a box plot (Figure 10).

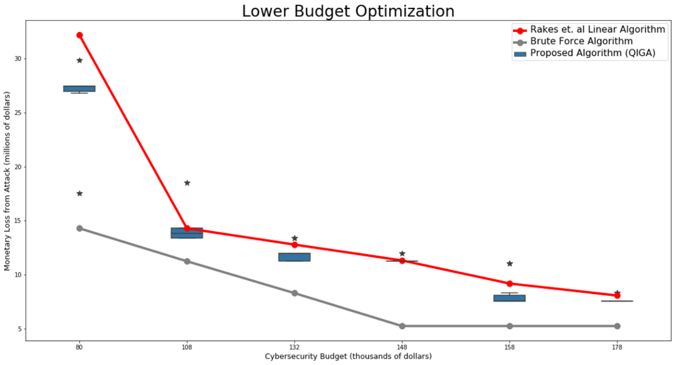
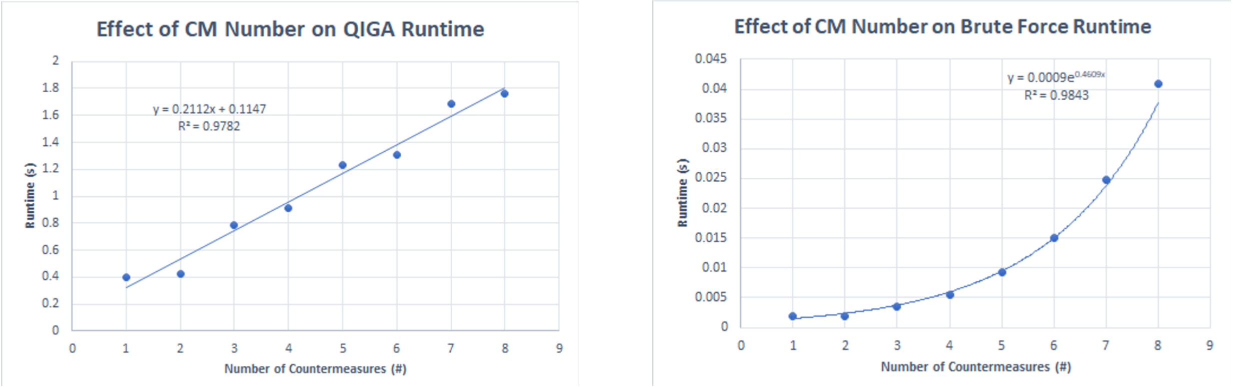


Figure 10: Distribution of Quantum Save's Results by Budget

When analyzing the box plot, data points that were outside of a standard 95% confidence interval, or a 1.57 IQR range were labeled as outliers (as can be seen by the stars on the box plot). So, based upon the distribution of outputs shown in the box plot, barring outliers, it appears as if Quantum Save's results for the expected-case number of attacks per year were either less than or equal to the linear optimization mode. [18]. Although in some budgets Quantum Save allocated budgets as effectively as the linear model, that still serves to emphasize the significance of Quantum Save in the realm of allocating cybersecurity budgets.

Time Complexity of Quantum Save

The second technique for analyzing the uniqueness in Quantum Save when compared to previous research, such as the linear optimization model proposed by Rakes et al. [18]. In typical countermeasure recommendation lists, there are approximately 50 to a 100 countermeasures meaning any algorithm that attempts to allocate cybersecurity budgets needs to scale well. Using an algorithm which scales exponentially would be extremely time consuming, which is a luxury that most smaller-medium targets can not afford. Instead, a cybersecurity budget allocation algorithm needs to be able to produce high number countermeasure recommendation lists in a reasonable amount of time. To ensure that Quantum Save could meet that criteria, the run time of the algorithm was measured. The data was then analyzed by finding both an experimental and theoretical relationship between the number of countermeasures in the recommendation list and the run time of the algorithm. When considering experimental data, the run time of both a traditional brute force algorithm and Quantum Save was charted. Brute force was chosen as a comparison algorithm because it has the best accuracy of any potential cybersecurity budget allocation problem.



(a) Quantum Save Time Complexity

(b) Brute Force Time Complexity

Figure 11: Time Complexity of Quantum Save vs. Brute Force

The run time results, which showed the relationship between the run time of a Brute Force algorithm and the size of the input, was fitted with an exponential trend line and an r-squared value of 0.9843 was obtained. This high r-squared value suggests that 98.43% of the variation in the run time of the algorithm was explained by the difference in input size, or the difference in the size of the countermeasure recommendation lists. The experimental results makes sense since Brute Force algorithms generally scale exponentially. This replication of Brute Force scalability provides justification for using this experimental method of determining run time. The other pair of results results, which showed the relationship between Quantum Save's run time and the size of the input, was fitted with a linear trend line and an r-squared value of 0.9782 was obtained. Again, this high r-squared value suggests that 97.82% of the variation in the run time of the algorithm was explained by the difference in input size, or the difference in the size of the countermeasure recommendation lists. This experimental data provides an initial suggestion that the algorithm scales linearly, but a theoretical justification is needed to supplement the idea.

Lemma 1. *Quantum Save can allocate budget, b, when choosing from countermeasures with input-size, n, in $\mathcal{O}(n)$ time complexity with oracles for quantum measurement of a single register.*

Proof. In the first line of Algorithm 2, the first phase of Quantum Save, the number of countermeasures is defined. This definition directly depends upon the input-size of the recommendation list, resulting in a complexity of $\mathcal{O}(n)$. From lines 2 - 4, basic variables for Quantum Save are defined. Since these variables are not correlated to the input-size, the resulting complexity is $\mathcal{O}(1)$. In line 5 - 6, the first loop begins which loops through the number of generations and the quantum register variable gets initialized. Again, since the number of generations is not correlated to the input-size, the resulting complexity of these lines is $\mathcal{O}(1)$. In line 7, a second loop begins which loops through every quantum register in the population. Since the quantum registers depend on the input size, these lines have a complexity of $\mathcal{O}(n)$. In lines 8 - 11, the quantum circuit is created and measured. Since the measurement of a single register does not depend on the size of the input, these lines have a complexity of $\mathcal{O}(1)$. Because the highest complexity is $\mathcal{O}(n)$ and there are no nested loops of higher complexity, the complexity of the first phase is $\mathcal{O}(n)$.

In the first two lines of Algorithm 3, the second phase of Quantum Save, the CISO variable is initialized and all of the CISO's are looped through. Since the number of CISO's does not vary based upon input size, the complexity of these lines is $\mathcal{O}(1)$. After a couple of definitions, which are $\mathcal{O}(1)$, the results of the quantum measurement are looped through in line 5. Since the job string would be longer if there are more countermeasures in the input, the complexity of this lines is $\mathcal{O}(n)$. Next, after a couple more definitions, which again are $\mathcal{O}(1)$, each countermeasure in the input array is looped through in line 9. Since the input array directly correlates with input-size, the complexity of this lines is $\mathcal{O}(n)$. Finally, in line

14, the maximum profit value produced is determined. Since the number of CISO remains constant, the complexity of this line is $\mathcal{O}(1)$. Because the highest complexity is $\mathcal{O}(n)$ and there are no nested loops of higher complexity, the complexity of the second phase is $\mathcal{O}(n)$.

After the definitions in the first two lines of Algorithm 4, which are $\mathcal{O}(1)$, the CISO's in the populationa re looped through. Since the number of CISO's does not vary based upon input size, the complexity of line 3 is $\mathcal{O}(1)$. Next, after a definition in line 4, line 5 of Quantum Save loops through every countermeasure in the binary array. Since the binary array increases with direct proportionality to the size of the inputs, the complexity of this line is $\mathcal{O}(n)$. Finally, the manipulation of the probability amplitudes, in lines 6 - 8, have a complexity class of $\mathcal{O}(1)$. Because the highest complexity is $\mathcal{O}(n)$ and there are no nested loops of higher complexity, the complexity of the third phase is $\mathcal{O}(n)$.

Since the complexity of all three phases are $\mathcal{O}(n)$, the final time complexity for Quantum Save is $\mathcal{O}(n)$. So, Quantum Save can allocate budget, b, when choosing from countermeasures with input-size, n, in $\mathcal{O}(n)$ time complexity with oracles for quantum measurement of a single register. □

Conclusion

The first part of this research was concentrated on developing an effective algorithm to optimize the knapsack problem. However, beyond that, this researcher has applied that algorithm to the real-world scenario of cybersecurity investment. From this research, an effective algorithm, Quantum Save, was developed on the IBM Q32 quantum simulator that increases the optimization of the knapsack problem. Due to the success of this algorithm in optimizing the knapsack problem, Quantum Save can be used for multiple practical applications. One application for Quantum Save is towards system design, which is a knapsack problem that can have major impact on the world as a whole. Essentially, system design process translates the customers' needs into a buildable system design that selects subsystems from an allowable set while remaining under a corporate budget. Chapman et al. [5] in an analysis of the system design process explain how system design problem is NP-complete similar to the Knapsack problem [5]. Both of these problems are attempting to maximize a certain value, either the required system performance or the amount of money in the knapsack, while remaining under a certain restraint, including a weight cap and a certain budget. Computer science is an ever-evolving field of study and finding the optimal solution to these fundamental problems, such as the knapsack problem, could be extremely beneficial especially as technology continues to grow.

Apart from the actual optimization of the knapsack problem, an accurate model for cyber security investment was proposed. Using Quantum Save, an optimal investment plan will be provided to the CISO so that his or her small-medium enterprise will be protected against the malicious attacks implemented by a hacker. The known vulnerabilities in the system will be handled and the company can perform more research and development without worrying that their database will be compromised. In this online network that society has woven in the modern world, the importance of keeping information safe cannot be emphasized enough. If a company has a bad security system implemented and their data is vulnerable to hackers from all around the world, the government and other business contractors will not want to do business. The reputation of these small-medium enterprises relies on the idea that their backend is secure enough so that the confidential data which will be entrusted to them doesn't get lost. With this cyber security investment plan, small-medium enterprises will have the means of strengthening their backend and increasing their reputation within society. In a modern wold where everyone's personal information is vulnerable, it is important for companies to invest in protecting that information and helping every person in the American society. Quantum Save can help small-medium compenies and hospitals save millions of dolalrs by effectively allocating cybersecurity budgets.

References

[1] "The Copenhagen and Many Worlds Interpretations of Quantum Mechanics." [Online]. Available: http://afriedman.org/AndysWebPage/BSJ/CopenhagenManyWorlds.html

[2] "Deloitte NASCIO Cybersecurity Survey | Deloitte Insights." [Online]. Available: https://www2.deloitte.com/us/en/insights/industry/public-sector/nascio-survey-government-cybersecurity-strategies.html

[3] A. Birkett, "How to Get Over Local Maximum? Everyone Hits It Eventually." [Online]. Available: https://cxl.com/blog/local-maximum/

[4] R. Bradley, T. Byrd, J. Pridmore, E. Thrasher, and R. Pratt, "An Empirical Examination of Antecedents and Consequences of Its Governance in US Hospitals," *Journal of Information Technology*, vol. 27, 2012.

[5] W. L. Chapman, J. Rozenblit, and A. T. Bahill, "System design is an NP-complete problem," *Systems Engineering*, vol. 4, no. 3, pp. 222–229, 2001. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.1018

[6] A. Fielder, E. Panaousis, P. Malacaria, C. Hankin, and F. Smeraldi, "Decision support approaches for cyber security investment," *Decision Support Systems*, vol. 86, pp. 13–23, Jun. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923616300239

[7] R. Forrester, "The Quantum Measurement Problem: Collapse of the Wave Function Explained," *Social Science Research Network*.

[8] A. Garg and S. Goyal, "Vector Sparse Representation of Color Image Using Quaternion Matrix Analysis based on Genetic Algorithm," 2017.

[9] S. Gootman, "OPM Hack: The Most Dangerous Threat to the Federal Government Today," *Journal of Applied Security Research*, vol. 11, pp. 517–525, 2016.

[10] H. Mia, M. A. Rahman, and M. Uddin, "E-Banking: Evolution, Status and Prospect," *Journal of ICMAB*, vol. 35, 2007.

[11] Y. Mirsky, T. Mahler, I. Shelef, and Y. Elovici, "CT-GAN: Malicious Tampering of 3D Medical Imagery using Deep Learning," *arXiv:1901.03597 [cs]*, Jun. 2019, arXiv: 1901.03597. [Online]. Available: http://arxiv.org/abs/1901.03597

[12] A. Montanaro, "Quantum algorithms: an overview," *npj Quantum Information*, vol. 2, no. 1, p. 15023, Nov. 2016, arXiv: 1511.04206. [Online]. Available: http://arxiv.org/abs/1511.04206

[13] C. Murawski and P. Bossaerts, “How Humans Solve Complex Problems: The Case of the Knapsack Problem,” *Scientific Reports*, vol. 6, no. 1, pp. 1–10, Oct. 2016. [Online]. Available: https://www.nature.com/articles/srep34851

[14] R. Nowotniak and J. Kucharski, “GPU-based tuning of quantum-inspired genetic algorithm for a combinatorial optimization problem,” *Bulletin of the Polish Academy of Sciences, Technical Sciences*, vol. 60, pp. 323–330, 2012.

[15] ——, “Higher-Order Quantum-Inspired Genetic Algorithms,” *IEEE*, 2014.

[16] X. Pan and T. Zhang, “Comparison and Analysis of Algorithms for the 0/1 Knapsack Problem,” *Journal of Physics: Conference Series*, vol. 1069, p. 012024, 2018.

[17] R. Rachwald, “Is banking online safer than banking on the corner?” *Computer Fraud & Security*, vol. 2008, pp. 11–12, 2008.

[18] T. Rakes, J. Deane, and L. Rees, “IT security planning under uncertainty for high-impact events,” *Omega*, vol. 40, pp. 79–88, 2012.

[19] M. Rocha and J. Neves, “Preventing Premature Convergence to Local Optima in Genetic Algorithms via Random Offspring Generation,” in *Multiple Approaches to Intelligent Systems*, 1999, pp. 127–136.

[20] A. Rohm and V. Swaminathan, “A Typology of Online Shoppers Based on Shopping Motivations,” *Journal of Business Research*, vol. 57, pp. 748–757, 2004.

[21] H. Wang, J. Liu, J. Zhi, and C. Fu, “The Improvement of Quantum Genetic Algorithm and Its Application on Function Optimization,” *Mathematical Problems in Engineering*, vol. 2013, p. 730749, May 2013. [Online]. Available: https://doi.org/10.1155/2013/730749