A Comparative Analysis of Population Annealing and Quantum Annealing solving Unit Commitment problem

Mohamad Altaweel
University of Stuttgart
mhd.altaweel98@gmail.com

Marco Aiello
Institute for Architecture of Application Systems
Service Computing
University of Stuttgart
marco.aiello@iaas.uni-stuttgart.de

***Abstract*—Unit Commitment is one of the fundamental mathematical optimization problems in electrical power management, which considers the coordination of power generation from a set of units to meet a target quantity of power. Many variants of the UC problem, which represents various real-life scenarios. UCP aims to minimize the total costs of power generation. That's why researchers are always driven to find effective methods to solve it. In this work, we compare between Population Annealing classical optimizing method and Quantum annealing optimization to solve the UCP in time needed to get a solution, studying the effect of four possible factors on the performance of the solvers. Overall, we see that population annealing is an effective method better than quantum annealing, but we still expect that quantum annealing should outperform the classical methods in the future.**

***Keywords*—Quantum Optimization, Unit Commitment, Energy power management, Quantum Annealing, Population Annealing**

1. Introduction

Electrical power management systems faces nowadays a lot of challenges due to the vast growth of consumption also in different advanced ways such as electric vehicles, IoT devices, smart cities...etc. These new technologies demand a stable power grids which can fulfil the dynamically changing need of consumption. In addition, a lot of interests in integrating of renewable energy sources into the power grids, which makes the grid structure more complex since the renewable sources impose new conditions and constraints to the grid architecture.

Unit Commitment problem (UCP) is one of the fundamental mathematical optimization problems in electrical power production, which considers the coordination of power production from a set of generators to achieve a specific level of power. With the rise of new challenges, there are many variants of the UC problem, which are often very difficult to solve. UC problem is considered as a very important problem since it aims to minimize the costs of power generation. Hence, hundreds of researchers has been made to find an effective method to solve it.

UCP is an NP-hard problem, which means there is still no algorithm that can solve it efficiently (polynomial function of time). Hence, it was a big challenge for researchers to solve this problem and use its application in reality.
The emergence of Quantum computers pushed the wheel forward in different domains with its ability to solve many of optimization problems thanks to the exponential speed-up it provides. However, the quantum hardware has still many restrictions which hampers the given size or complexity of the problem to solve on quantum computers. Researchers investigate in many studies how can the current quantum technologies solve the UCP and if it presents a potential solver in the future.

In this work, we compare the performance between classical and quantum methods to solve the UCP. By formulating UCP as Mixed-Integer Non-Linear Problem (MINLP), we can transform it into a Quadratic Unconstrained Binary Model optimization (QUBO), considering the constraints which are relevant for renewable energy sources case [1]. To solve this model, we apply it on Populated Annealing solver (Classical computing) provided by Microsoft Quantum Azure and on Quantum Annealing (Quantum computing) provided by Amazon brackets, which uses D-Wave systems. The experiment inspects the performances of both solvers regarding the change of time loads, number of generators, adding Spinning reserves which handles possible peaks, and adding renewable resources which has a dynamically changing power generation over time. The results show that the classical approach has still a better performance. However, we conclude that in the future quantum annealing can overcome the performance of classical approach as we see later by increasing the time loads.

This paper is structured as follows: In chapter 2, we present a brief explanation about quantum computing and how it can be used in optimization domain, comparing between both used methods quantum annealing and populated annealing.We review the latest works and literature focused on solving UCP with different approaches. We present our method in the chapter 3, showing how we modeled the problem, including the main constraints. The implementation vary depending on the used platform. The experiment to compare both approaches is explained in chapter 4.3.4. Finally, we discuss our work and analyze the limitations of it in chapter 5.

2. Background

We provide a brief explanation in this section about the concepts of quantum computing and optimizations used in this work. Moreover, we review the recent literature looking into solving the UCP with different methods.

2.1. Foundations

Quantum computing is a new computation technology that applies the quantum mechanics instead of classical physic representation of information as in classical computers. Quantum computers are able to solve any problem a classical computer can solve. Thus, it follows the Church–Turing thesis which means that they do not introduce any new definition of computable functions but only provides efficient performance over classical computers [2]. We introduce the main concepts of quantum mechanics applied in quantum computing to understand the process of optimization applied lately.

2.1.1. Quantum mechanics. In classical physics, a bit or any object can be in one state at the same time, mainly ”exists” or ”does not exist”. Quantum mechanics goes more behind this simple rule and can describe more complex case on atom and electrons level.

quantum states. A quantum state of an object describes the probability distribution of all possible actual states that this object might have. A logical bit has a range of two values ”0” or ”1”. Thus, its quantum state describes the probability distribution when this **(Qu)**-bit would have the value ”0” or ”1”.

superposition. This principle contradict the fact that an object can be at one of its state at a specific time. Superposition states that any two or more quantum states can be added together which still gives another valid quantum states. This means that a Qubit can hold both values together (superposed) as not in classical bit case. Superposition is the secret behind of the exponential speed up of quantum technologies since it allows processing all possible configuration in a parallel way at the same time.

entanglement. Entanglement as defined per J.D. Hidary [2] that the states of the two qubit systems are not separable. which means that both qubit are correlated with each other and any change of the state of one system would affect the state of the other one. This allows us to link between variables and restrict the possible configuration that can come from the combination of both systems.

measurement. The measurement in its classical definition mean to read a current value of an object's property such as the temperature, speed, or frequency. This act has no effect on the state of the object but not in quantum space. Acting a measurement on a Qubit destroys its superposition and transmit it into a classical state of one the possible configuration it could have. Moreover, this process is not revertible, which means we can not restore the previous state before the measurement [2].
In superposition state, we are not able to get values from the Qubit.That's why we need to perform a measurement in order to get the final results which gives one of the possible values according to the probability distribution of the Qubit itself. Quantum algorithm tries to amplify the probability of the right or optimal solution so when we ensure to get the optimal results by measuring in the end of the computation.

2.1.2. Quantum Annealing. Quantum Annealing was introduced as a powerful optimization method which integrates the quantum fluctuations into the simulated annealing process of optimization problems. [3] This technique is inspired by classical simulated annealing, which is an approximation method for global minimum of an optimization problem based on the technique involving heating and controlled cooling of a material to alter its physical properties [4]. These techniques are when the search space is discrete, and we are trying to approximate the global optimum instead of using classical algorithms such as brute-force or branch-and-bound.
Quantum annealing use the concept of low-energy states search of a problem, which means the optimal solution of a problem. A qubit - which represents a variable in a problem - starts with its superposition state. The annealing process raises a small summit between 0 and 1 classical states so that the quantum has an equal probability of ending up in one of those states. To control this probability, we add a bias to the qubit which is physically an external magnetic field and the qubit minimizes the energy which is then increasing the probability of failing into one of the states as shown in the example of figure 1. Moreover, to link the variables or qubits in this context, we use a coupler which relies on the quantum physics phenomenon ”entanglement”. A coupler can force both qubits to end in the same state (either 00 or 11) and the correlation weights between qubits can be set by the user as bias [5].

briefly, to solve an optimization problem, we start with a set of qubits in superposition states. During the annealing, the system push the defined bias and couplers which turns the system into an entangled state. In the end, each qubit should have a classical state that represents the minimum energy state of the problem, or one very close to it. The process takes only microseconds to finish. [5]

2.1.3. Population Annealing. The population annealing algorithm is based on simulated annealing and shares features

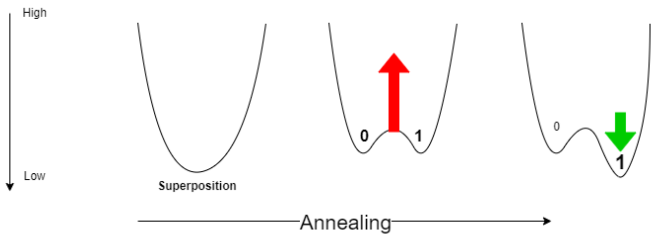

Figure 1. Quantum Annealing physical process

with other techniques such parallel tempering [6], histogram re-weighting [7] and diffusion Monte Carlo [8]. In a similar way to simulated annealing, it is a single pass algorithm with an annealing schedule. A set of replicas(population) of the system is cooled depending on the annealing schedule. The main difference to classical simulated annealing is that the population is always maintained in equilibrium throughout the whole process at every temperature [9].
Intuitively, Population Annealing starts with a set of walkers(solvers) that floats in the space of possible configurations of the problem. Each walker can explore the own neighborhood and once it finds a better state, then the rest of the population is pulled into it. To solve an optimization problem, we start with a set of walkers in random states in high temperatures. This allows an easier transition to the lower-energy states, preferring the low-cost move from any walker. When we reach lower temperatures, the Metropolis Criterion strongly tends to transition which decrease the cost and re-sampling can eliminate non-competitive states. Finally, population is ensemble into the same lowest-cost state discovered [10].

2.2. State-of-the-Art

Many researchers are interested in using the quantum technologies. One of the latest researches [11] used Quantum Approximation Optimization Algorithm (QAOA) to solve the simple model of Unit commitment without any constraints and states that classical solver still outperforms quantum ones for small systems. However, the simulation results ensures that classical solver is not capable to solver larger instances of the problem which concludes that quantum technology is the one can solve the problem with the presence of capable quantum hardware. This paper was based on previous work [12], which used Quantum Annealing, as in our case, to solve the simple UC. To prevent discretizations costs, since QA method can deal only with discrete models, they used QAOA to keep the model in its continuous form. Both works handle the simple scenario of UCP without any constraints. We go beyond the simple model of UC and adds more of them, especially those related to renewable energy sources.

Due to the current hardware limitations of quantum computers, researchers are looking for alternative ways to benefit from quantum computing with its current restrictions. In this paper [13], authors proposed a new method which decompose the UCP and make use of coordination-supported framework to solve the large-scale instance of UCP. The distributed version divides the problem into several micro-grids, so each one optimizes its objective function and only the joint variables are shared to get the updated multiplier using Quantum Alternate Direction Method of Multiplier (QADMM). The proposed technique is very promising and authors tested it on three different scenarios. The paper states only the validity of the solution, which can be compared with the outcomes of classical solvers. However, we provide a detailed measurement of time on our experiments to be able to compare with in the same conditions and given model with classical solvers. In addition, we investigate of the possible effect of four factors on the performance of the solvers which guides us in which direction we should optimize our grid architecture.

Another work [14] tend to corporate between classical and quantum solvers in a hybrid way to overcome the quantum limitations. The UC problem is decomposed into three subproblems: a quadratic subproblem, a quadratic unconstrained binary optimization (QUBO) subproblem, and an unconstrained quadratic subproblem. The QAOA algorithm is used to solve the QUBO subproblem, while the classical solver can handle the other parts.Eventually, the three subproblems are then coordinated iteratively using a three-block alternating direction method of multipliers algorithm similar to work [13]. Simulations done in this work states that the performance of the proposed decomposition algorithm such that the results of the problem for various loads match the results of a centralized classical solver.

In summary, the current bounded hardware specifications of quantum computers are still forming an obstacle to fully benefit from it. Hence, researchers intend to decentralize the problem to make it compatible with the quantum technology. Experiments from previous work states that quantum systems can not fully overcome the classical solvers in small scenarios.

3. Method

We introduce the mathematical form of UC in this section, then we explain the concepts to implement it to submit as a problem for Population Annealing solving as well as for Quantum Annealing.

3.1. Formulation of Unit Commitment

We formulate UCP as an MINLP. Our aim is to minimize the cost function, which is represented as following equation:

$$\min_{u,p} \sum_{i,t} u_{i,t}(A_i + B_i p_{i,t} + C_i p_{i,t}^2) + s_{i,t} \tag{1}$$

i is the index of the power generator.
t is the time unit.
$u_{i,t}$ is an indicator whether the power generator i is on/off at time t.
$p_{i,t}$ is the power level produced by generator i at time t.
A_i, B_i, C_i are fixed costs for each generator i.
$s_{i,t}$ represents the startup/shutdown costs defined as follows:

$$s_{i,t} = \begin{cases} A_i^U \text{Up cost} & \text{if } u_{i,t-1} < u_{i,t} \\ 0 & \text{if } u_{i,t-1} = u_{i,t} \\ A_i^D \text{Down cost} & \text{if } u_{i,t-1} > u_{i,t} \end{cases} \tag{2}$$

In addition, the solution should preserve a set of constraints which are related physical and system operational conditions [15]. In review [1] authors defined the set of constraints which are the set of later evolution of RES-UC models. We include these constraints to our model, which we intend to solve:

Power limit constraints. Each unit has a minimum and maximum output it can produce. For each time t the generated power should be in the range of the Unit capacity.

$$P^i_{min} < p_{i,t} < P^i_{max} \tag{3}$$

Power balance constraints. All active units at time t must satisfy the consumer demand at this time. It means that the sum of all generated power must be equal the consumer demand:

$$\sum_{i=1}^{N} p_{i,t} = D_t \quad \forall t \tag{4}$$

Minimum Up-time Constraint. is the minimum time that a unit i should be active before it is turned off again

$$T_i^{on} > MUT_i \tag{5}$$

In other words, the sum of the indicators over all possible ranges from $t = 1$ to $t = T - MUT$ should be more than the defined minimum uptime of a unit i only after it is turned on at a specific point of time.

$$\sum_{t=1}^{MUT} u_{i,t} > MUT_i \tag{6}$$

Minimum down-time constraint. Similarly. when a unit i is turned off, it can not be turned on again after a piece of time

$$T_i^{off} > MDT_i \tag{7}$$

which we can represent using indicators sum as in (6) as follows:

$$\sum_{t=1}^{MDT} u_{i,t} = 0 \tag{8}$$

Ramp rate up/down constraint. The change of output power can not increase or decrease directly. any change of production is restricted by a defined ramp rate limits.

$$P_{i,t-1} - P_{i,t} \leq UR_i \quad \text{if change increases} \tag{9}$$

$$P_{i,t} - P_{i,t-1} \leq DR_i \quad \text{if change decreases} \tag{10}$$

Spinning reserves constraint. Spinning Reserves (SR) is an extra consuming demand of power that is required to handle an expected peak or the loss of the most loaded unit at time t.The formulation is defined in [16], [17]:

$$\sum_{i=1}^{N} p_{i,t} \geq D_t + SR_t \quad \forall t \tag{11}$$

We notice that this constraint already covers the power balance. but sometimes if the solution was not capable of generating a power reserves for further peaks, then it must at least fulfill the demand at that time.

Must run units. A number of units could be specified by the operator which defines how many unit must at least be active at each time. We define this constraint as the following equation:

$$\sum_{i=1}^{N} u_{i,t} \geq MRU_t \quad \forall t \tag{12}$$

However, solvers accept the problems in Quadratic Unconstrained Binary Optimization form. Hence, we need to modify our model to a QUBO form before start implementing.

3.2. Quadratic Unconstrained Binary optimization(QUBO)

Binary Optimization is a type of optimization problems where the set of variables are restricted mainly to two values, 0 or 1 subject to a set of constraints. Polynomial Unconstrained Binary optimization (PUBO) is a special case where there are no constraints and the objective function is in polynomial form [18]. When the order of the polynomial is 2, then we deal with a Quadratic model, mainly QUBO. The motivation behind transformation to QUBO form is that it is equivalent to Ising model [19] which can be solved using Quantum Annealing method.

3.2.1. discretize the model. QUBO is a discrete model which means all variables can only have integer values. Hence, we discretize the power level output range into a vector starting from 0 as the unit turned off, the minimum output power and then increment by 1 for each following element till we reach the maximum power output.

$$\begin{pmatrix} 0 \\ P^i_{min} \\ P^i_{min} + 1 \\ P^i_{min} + 2 \\ \cdot \\ \cdot \\ P^i_{max} \end{pmatrix}$$

In this case, we cover the power limit constraint 3.

3.2.2. Define the variables. QUBO variables are binary variables. That's why we define the indicators $u_{i,t}$ as variables which the solver try to optimize the problem by controlling its values. To include the power level values, we expand the indicators from two dimensions, i the generator index and t the time unit, to three dimensions i, p, t. In this case, the variable $u_{i,t,p}$ indicate if the generator i produces power quantity q at time t. Figure 2 shows an example for each time unit t that the optimizer should find of a suitable configuration of power levels like playing on a keyboard. Another point is that solvers consider multiplying the variable with itself as the multiplication of two different variables. That means in equation (1) we have to multiply three different variables: $u_{i,t}$ and $p_{i,t}^2$. In this case, we rise the order of polynomial to 3 and the model is not in QUBO form anymore.

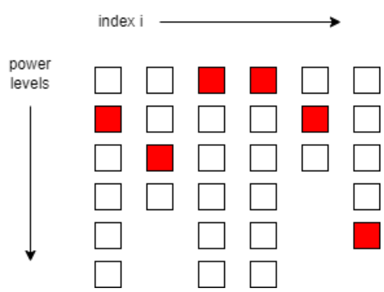

Figure 2. at time t, solver has possible combinations of power-levels of units which should select, where selecting two options in the same column is not possible

3.2.3. Inserting constraints into the objective function. QUBO model does not include any constraints subject to the variables. Hence, we should merge the defined constraints into the objective function by summing them with the cost function so that the solver can optimize the whole equation and find a minimum value which can respect the constraints in the same time. Equation (2) we move the demand to the other side and raise it to the power order 2 to force the optimizer to minimize the value to 0. Similarly, we apply it to equation (8) so we got the following polynomials:

$$(\sum_{i=1}^{N} p_{i,t} - D_t)^2 \quad \forall t \tag{13}$$

$$(\sum_{t=1}^{MDT} u_{i,t})^2 \tag{14}$$

However, we have some inequation constraints that could not fit to the function directly so we need to reform them in the next part 3.2.4.

3.2.4. Turn an inequality to an equation. In order to include an inequation constraint into the object function (1), we need to reform it into an equation. In general, if we have the inequation $x \geq y$ defined only with integer values, then x could have all values starting from $y+1$. by moving all variables to one side we have $x+y \geq 0$, we can optimize the sum $x+y$ to have the value of 0. Hence, our optimization function would be $(x+y)^2$. If the optimizer can not find a solution where the sum is 0, it is still acceptable to have values greater than 0 since it does not violate the original inequation. To force the system to try to keep the $>$ and not only land in the equal $=$ case. so the final equation would be $(x+y-1)^2$ which optimizing it ensure the validity of the inequation $x \geq y$. We apply this trick to all inequations (6,9,10,11,12) as follows:

$$(\sum_{t=1}^{MUT} u_{i,t} - MUT_i - 1)^2 \tag{15}$$

$$(P_{i,t-1} - P_{i,t} - UR_i - 1)^2 \tag{16}$$

$$(P_{i,t} - P_{i,t-1} - DR_i - 1)^2 \tag{17}$$

$$(\sum_{i=1}^{N} p_{i,t} - D_t - SR_t - 1)^2 \tag{18}$$

$$(\sum_{i=1}^{N} u_{i,t} - MRU_t - 1)^2 \tag{19}$$

3.3. Implementation

We aim to implement a highly reliable software which can be used to test different data sources or apply it to other optimizer. In addition, we can build a service which can be online deployed and integrated with different business cases. That's why we followed the separation of concerns [20] principle in our implementation.
Figure 3 clarifies the architecture of the system. It composes of three parts: The first component reads the records and values from the data source to save it to an abstract model of the grid. This model is a network of thermal and renewable energy resource.i.e in our case a wind turbine. Each generator saves necessary information which are mentioned previously in the objective function such as power levels, ramp-up/down rate, minimum uptime ...etc. The renewable energy generator is a special case because it may have different power limits for each time depending on the physical conditions, in contrast to thermal generators which has always a fixed power limits over the time. This optimizer endpoint part can then access this information from the model to hand them over to the optimizer.
In this case we can benefit from different data sources without any need to change the optimization function part and use other optimizations for the same data source.

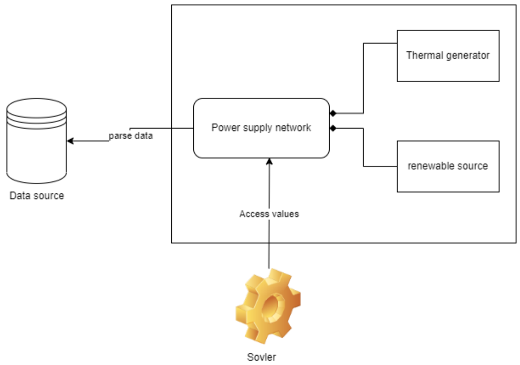

Figure 3. The coarse architecture of implementation

3.3.1. Microsoft Azure Quantum. Azure Quantum is a cloud service part of azure cloud which enables users to use different quantum and optimization solutions. With use of Q programming languages, we can submit optimization problems to different available optimizers. Solving a problem can be done by submitting an object of type "Problem" which contains a set of "Term" or "Slcterm" objects. A term is variable with a specific index and assigned coefficient integer which is in our case the indicator u_{itp} which is multiplied with some numbers regarding which part of the objective function. Furthermore, "Slcterm" is squared linear combination of grouped terms which simplify adding the reformed constraints (6,9,10,11,12,2,8). Azure Quantum allows the access of different solvers such as Parallel Tempering, Simulated Annealing, and Population Annealing. However, Available quantum technologies can accept only quantum circuits with 10 or 11 qubits at maximum [21]. That's why we use Amazon brackets in the next section 3.3.2.

3.3.2. Amazon bracket & D-Wave systems. Amazon bracket is also a cloud service which allows access to different quantum technologies such as D-Wave 2000Q and D-Wave Advantage. With bracket library integrated to python programming language we can submit the problem which is modelled as a dictionary and List, where each key is a tuple of the index and its value is the coefficient for all quadratic parts of the function and the index in list is the variable index with its value as a coefficient. The constant part is ignored here. We submit the problem to solve by D-Wave Advantage which has up to more than 5000 available Qubits and 15 couplers per Qubit [22].

4. Experiment

To evaluate the performance of both optimizers, we designed a comparative experiment between the two of them, in which we input the same data and compare the timing performance in milliseconds.

4.1. data acquisition

The data is acquired from Power Grid Lib - Unit Commitment repository on github [23] which obtained originally from three different sources [24], [25], [26]. The data is saved in JSON format where each file contains the demands as List of length T associated with all generators. Each generator contains the necessary information mentioned in the Unit Commitment model.
However, the production cost here is provided as one value not as three different coefficients A, B, C. Hence, we multiplied the cost directly with the power level and the indicator variable $u_{i,t,p}$. Also, we were not able to retrieve the shutdown costs, so we assumed that shutting down the unit has no extra costs to consider. Renewable energy source generators does not include any costs, that's why we assumed here that there is no operational costs for those units.

4.2. Experiment design

In order to analyze the performance of each solver, we designed an experiment to execute with four potential factors, that has an impact on the complexity of the UCP instance:

- Increasing the demands (Loads) : adding a new demand which extends the whole given time T means to expand all the variables for a new time unit. This enlarges the size of the given objective problem massively.
- Adding Spinning Reserves: the spinning reserve requirement sharpen the problem constraint by forcing the grid to produce more power at each time.
- Increasing power plants: adding new generators allows having more combinations which might give more optimal solutions. However, having more possible configurations makes it harder to reach the optimum solution fast and would worsen the performance.
- Adding renewable energy sources: as mentioned previously, RES generators have different power limits each time. This means that we have more possible configurations but in a complex way since each time we consider another limits for those generators. This represents a challenge to UCP solver because it impacts the internal structure of the problem and could make it harder to plan the power supply process.

To test each factor, we hold the other factors' values and inspect the performance while changing the currently tested factors.
We compare the time need to solve the given problem of the two solver. We record the "Execution time" of population annealing, which means the time the CPU started to solve the problem. On the Quantum Annealing, we get the QPU Access time, which is the time to execute a single Quantum Machine Instructions(QMI) on a QPU. The access time is represented in the following equation:

$$T = T_p + \Delta + T_s$$

where T_p is the programming time, T_s is the sampling time, and Δ is an initialization time spent in low-level operations (10-20 ms for Advantage systems) [5].

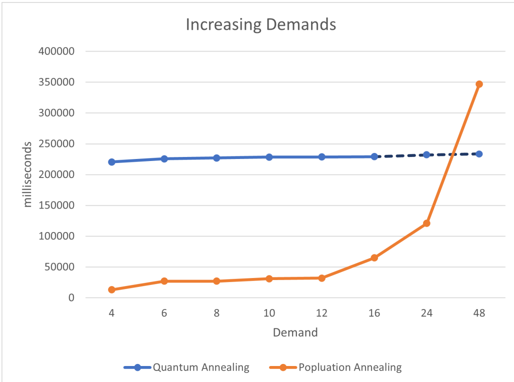

Figure 4. Increasing the demands has a massive impact on the performance

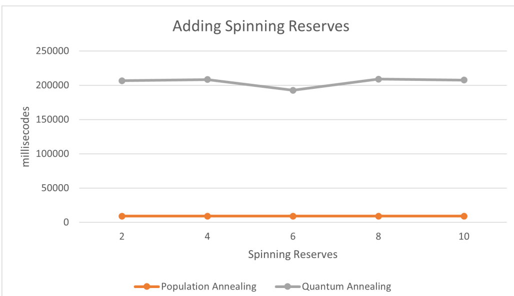

Figure 5. Adding spinning reserves has nearly no effect on the performance

4.3. Experiment results

Due to the quantum hardware limits by coupling the qubits with each other, we conduct the experiment on relatively low number of power units.
Overall, the classical solver still outperforms the quantum solver, which matches the result found in [11]. However, we expect that by larger instances the quantum will outperform the classical solver, since the classical solver showed an exponential performance by increasing the time loads. We discuss the results in details in the following parts.

4.3.1. Increasing Demands. we start with only four demands ($T = 4$) and start increasing the demands, which means extending the time horizon. At a specific point of time ($T = 12$), the solving time needed started to grow up exponentially for population annealing, while it stays the same for quantum. That's why we expect that the quantum solver is able to overcome when we have more than 24 demands, $T = 24$ as marked in figure 4

4.3.2. Adding spinning reserves. Pushing the power grid to produce more power does not affect the solving time needed. As we see in figure 5, we added spinning reserves for two time of units then increase it gradually for the whole time, and it shows no impact on the solvers' performance.

4.3.3. increasing power plants. We notice in figure 6 a small growth jump of time needed to solve the problem on both solvers. However, there is no further impact when we increase the number of power units.

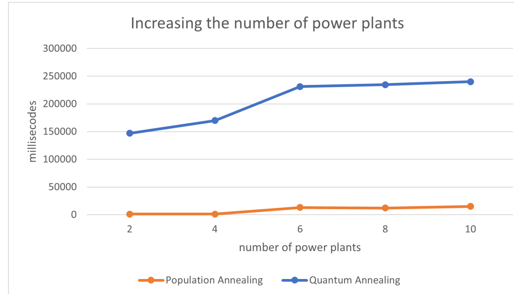

Figure 6. Increasing power units does not complicate the solving process.

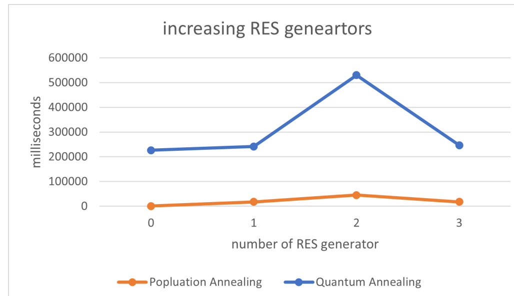

Figure 7. Increasing the number of RES generator included in the grid.

4.3.4. increasing renewable energy sources. Including RES power units showed an interesting case. The classical solver is always the better solver, needed less time than the quantum solver. Both solvers needed more time to solve the UCP included 2 RES units and 4 normal units than a ratio of 3:3. It means that the problem instance is more complex, with a ratio of 2:4. We repeated the experiment with different data, and it showed the same behavior. Logically, it should be equal or harder when increasing the RES units, but not easier. This case derives the power management optimization to investigate, as the ratio of RES units should be included in the power grid.

5. Conclusion

In this paper, we investigate of proper methods to solve the Unit Commitment problem, which is a fundamental optimization problem in power management domain.
The arising of quantum computers encourages researchers to solve the UCP efficiently, since there are a lot of methods based on classical algorithms trying to find an approximate solution for it.
We compare between Population Annealing classical optimizing method and Quantum annealing to solve the UCP in time needed to get a solution, studying the effect of four possible factors on the performance of the solvers. Overall, we see that population annealing is an effective method better than quantum annealing for small size problems, which

we are capable to apply now on both hardware products. It is still expected that quantum annealing should outperform the classical methods when the hardware structure is improved to process larger problems.

Limitations and future works

We outline the main critical points to our work here for further researches to discuss and investigate:

- The Population Annealing outperforms the quantum annealing on small instances of UCP. However, we expect that on larger problem the quantum technology should exceed the classical solving methods.
- The quality of returned solutions must also be analyzed, comparing with the time needed to solve the problem. Quality and Validity of solutions are important benchmarks for provided solutions. In this work, we focused only on time needed to solve the problem.
- Adding RES power generators showed an interesting case. The experiment revealed that for a specific ration 2:4 takes more time to solve than 3:3 which is expected to be harder. A deep investigation should justify the behavior of solvers to get a clear statement which would play an important role in different business cases in power management and supply industry.

References

[1] S. Y. Abujarad, M. Mustafa, and J. Jamian, "Recent approaches of unit commitment in the presence of intermittent renewable energy resources: A review," *Renewable and Sustainable Energy Reviews*, vol. 70, pp. 215–223, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032116310140

[2] J. D. Hidary, "Quantum computing: An applied approach," *Quantum Computing: An Applied Approach*, 2019.

[3] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse ising model," *Phys. Rev. E*, vol. 58, pp. 5355–5363, Nov 1998. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.58.5355

[4] P. Siarry, *Simulated Annealing*. Cham: Springer International Publishing, 2016, pp. 19–50. [Online]. Available: https://doi.org/10.1007/978-3-319-45403-0_2

[5] "What is quantum annealing?" https://docs.dwavesys.com/docs/latest/c_gs_2.html, 2022.

[6] M. Sambridge, "A Parallel Tempering algorithm for probabilistic sampling and multimodal optimization," *Geophysical Journal International*, vol. 196, no. 1, pp. 357–374, 10 2013. [Online]. Available: https://doi.org/10.1093/gji/ggt342

[7] A. Z. Panagiotopoulos, "Monte carlo methods for phase equilibria of fluids," *Journal of Physics: Condensed Matter*, vol. 12, no. 3, pp. R25–R52, dec 1999. [Online]. Available: https://doi.org/10.1088/0953-8984/12/3/201

[8] P. J. Reynolds, J. Tobochnik, and H. Gould, "Diffusion quantum monte carlo," *Computers in Physics*, vol. 4, no. 6, pp. 662–668, 1990. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/1.4822960

[9] J. Machta, "Population annealing with weighted averages: A monte carlo method for rough free-energy landscapes," *Physical Review E*, vol. 82, no. 2, aug 2010. [Online]. Available: https://doi.org/10.1103%2Fphysreve.82.026704

[10] "Population annealing," https://docs.microsoft.com/en-us/azure/quantum/optimization-population-annealing, 04/19/2022.

[11] S. Koretsky, P. Gokhale, J. M. Baker, J. Viszlai, H. Zheng, N. Gurung, R. Burg, E. A. Paaso, A. Khodaei, R. Eskandarpour, and F. T. Chong, "Adapting quantum approximation optimization algorithm (qaoa) for unit commitment," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2021, pp. 181–187.

[12] A. Ajagekar and F. You, "Quantum computing for energy systems optimization: Challenges and opportunities," *Energy*, vol. 179, pp. 76–89, jul 2019. [Online]. Available: https://doi.org/10.1016%2Fj.energy.2019.04.186

[13] N. Nikmehr, P. Zhang, and M. Bragin, "Quantum distributed unit commitment," *IEEE Transactions on Power Systems*, pp. 1–1, 2022.

[14] R. Mahroo and A. Kargarian, "Hybrid quantum-classical unit commitment," in *2022 IEEE Texas Power and Energy Conference (TPEC)*, 2022, pp. 1–5.

[15] S. Sen and D. Kothari, "Optimal thermal generating unit commitment of large power system: a novel approach," in *Proceedings of IEEE TENCON '98. IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control (Cat. No.98CH36229)*, vol. 2, 1998, pp. 474–478 vol.2.

[16] B. Gjorgiev, D. Kančev, M. Čepin, and A. Volkanovski, "Multi-objective unit commitment with introduction of a methodology for probabilistic assessment of generating capacities availability," *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 236–249, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197614002346

[17] K. Chandrasekaran, S. Hemamalini, S. P. Simon, and N. P. Padhy, "Thermal unit commitment using binary/real coded artificial bee colony algorithm," *Electric Power Systems Research*, vol. 84, no. 1, pp. 109–119, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779611002471

[18] F. Glover, J.-K. Hao, and G. Kochenberger, "Polynomial unconstrained binary optimisation–part 1," *International Journal of Metaheuristics*, vol. 1, no. 3, pp. 232–256, 2011.

[19] A. Lucas, "Ising formulations of many NP problems," *Frontiers in Physics*, vol. 2, 2014. [Online]. Available: https://doi.org/10.3389%2Ffphy.2014.00005

[20] P. A. Laplante, *What Every Engineer Should Know about Software Engineering (What Every Engineer Should Know)*. USA: CRC Press, Inc., 2007.

[21] "Azure quantum computing," https://docs.microsoft.com/en-us/azure/quantum/, 2022.

[22] "Amazon braket," https://aws.amazon.com/de/braket/, 2022.

[23] "Power grid lib - unit commitment v19.08," 2022.

[24] C. Barrows, A. Bloom, A. Ehlen, J. Ikäheimo, J. Jorgenson, D. Krishnamurthy, J. Lau, B. McBennett, M. O'Connell, E. Preston, A. Staid, G. Stephen, and J.-P. Watson, "The ieee reliability test system: A proposed 2019 update," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 119–127, 2020.

[25] M. H. Krall, Eric and R. P. O'Neill, "Rto unit commitment test system," *Federal Energy Regulatory Commission*, 2012.

[26] B. Knueven, J. Ostrowski, and J.-P. Watson, "On Mixed-Integer Programming Formulations for the Unit Commitment Problem," *INFORMS Journal on Computing*, vol. 32, no. 4, pp. 857–876, October 2020. [Online]. Available: https://ideas.repec.org/a/inm/orijoc/v32y4i2020p857-876.html