Technical report Group 8

Alberto Bottarelli, Oriel Kiss
(Dated: March 2025)

I. TECHNICAL DESCRIPTION

A. Problem

We treat the problem of bi-directional EV charging of N electric vehicles (EVs) over T time steps of uniform lenght Δt. In order to do this, we need $N \times T$ binary variables $z_{n,t}$. Each one of these represents the charging state of vehicle n at time step t, with $n \in \{0, ..., N-1\}$ and $t \in \{0, ..., T-1\}$. Assuming that the vehicles are allowed to charge and sell energy to the grid, the problem can be expressed as the task of computing the ground state of the following Hamiltonian:

$$H_c = \sum_{t=1}^{T} \sum_{n=1}^{N} \Delta t P_n^0 \Big(c_t \sigma_{n,t}^z + \delta_t (1 - n_{n,t}) \Big) = \sum_{t=1}^{T} \sum_{n=1}^{N} n_{n,t}(2c_t - \delta_t) - c_t + \delta_t \,, \tag{1}$$

where the eigenstates $|0\rangle, |1\rangle$ of each Pauli matrix $\sigma_{n,t}^z$ represent the charging ($|0\rangle$) or discharging($|1\rangle$) of the vehicle.

The parameters represent the following quantities:

- P_n^0 = Charging power of the electric vehicles. In principle, it can be vehicle dependent. We will assume it to be equal for each vehicle.
- c_t = price for buying energy from the grid at timestep t
- δ_t = price for selling energy to the grid at timestep t
- Δt: Lenght of the time step of the charging process.

In order to make the problem non-trivial, some constraints need to be taken into consideration. We consider the following ones:

$$\sum_{t=1}^{T} P_n^0 \sigma_{n,t}^z = \tilde{E}_n \;\; \forall \;\; n \tag{2}$$

$$\sum_{n=1}^{N} P_n^0 \sigma_{n,t}^z = P \;\; \forall \;\; t \tag{3}$$

$$\tag{4}$$

where $\tilde{E}_n$ and P are the respectively the target energy level of each vehicle and the constant flow of energy from the grid. These constraint ensure that each vehicle reaches the target energy level at the end of the charging schedule. And that the grid is always at a constant output in order to avoid being stressed. Given that these are linear equality constraint, they can be imposed on the problem by adding the following quadratic energy penalty to the Hamiltonian (we only show the penalty for the first set of constraints as the second set is obtained in the same way.

$$H_P = \lambda \sum_n \left(\sum_{t=1}^{T} P_n^0 \sigma_{n,t}^z - \tilde{E}_n \right)^2 = \tag{5}$$

$$\lambda \sum_n \left((P_n^0)^2 \sum_{t,t'=1}^{T} \sigma_{n,t}^z \sigma_{n,t'}^z - 2\tilde{E}_n P_n^0 \sum_{t=1}^{T} \sigma_{n,t}^z + \tilde{E}_n^2 \right). \tag{6}$$

Here, λ is an energy penalty term with the role of shifting the energy levels of the unfeasible states, such that the remaining low energy states represent only feasible configurations. $\tilde{E}_n$ is the target charge level for vehicle n.

Reshuffling the terms and expressing them using number operators, merging the cost and the penalties for the constraints, we get as final Hamiltonian the following expression:

$$H = H_c + H_P = \sum_{n,t} n_{n,t} n_{n,t} (2c_t - \delta_t - 4\lambda \tilde{E}_n P_n^0 - 4\lambda (P_n^0)^2) + \sum_n \sum_{t,t'} \lambda (P_n^0)^2 n_{n,t} n_{n,t'} + \tag{7}$$

$$+ \sum_{n,t} 2\lambda P_n^0 \tilde{E}_n + \delta_t - c_t - \sum_n \sum_{t,t'} \lambda (P_n^0)^2 + \sum_n \tilde{E}_n^2 \tag{8}$$

where the first line represents entries of the QUBO matrix and the second represents a constant shift in the energy values(which will be ignored).

B. Quantum Algorithm description

The cost Hamiltonian corresponds to a classical cost function that can be expressed in QUBO form

$$C(z) = \mathbf{z}^T Q \mathbf{z}\,, \tag{9}$$

Where the matrix Q represents the QUBO matrix and the string $\mathbf{z}$ represent the binary configurations (i.e. the quantum states). Simulating Pasqal architecture, we study this using the adiabatic algorithm.

In particular, we consider the analog Hamiltonian

$$H_Q = \sum_{i=1}^{N} \frac{\hbar \Omega(t)}{2} \sigma_i^x - \sum_{i=1}^{N} \frac{\hbar \delta(t)}{2} \sigma_i^z + \sum_{i<j} \frac{C_6}{|r_i - r_j|^6} n_i n_j \tag{10}$$

and map the QUBO matrix that defines our problem into the coefficients $\frac{C_6}{|r_i - r_j|^6} n_i n_j$. We employ machine learning in order to train the adiabatic schedule in order to find the optimal one for this problem. We then perform a noise study to understand the resilience of the found schedule to depolarization and SPAM errors. We quantify the performance of the algorithm by computing the approximation ratio of the found solution.

1. *Quantum annealing*

Adiabatic State Preparation (ASP) has been developed as a technique to prepare the ground state of an arbitrary target Hamiltonian [1, 2]. The protocol begins with the system initialized in a ground state that is easily prepared and then gradually evolves towards the target Hamiltonian over a finite time interval. Under the assumption of sufficiently slow evolution, the final state should correspond exactly to the desired ground state. However, in practical implementations, the evolution time is finite, and therefore, perfect preparation cannot be guaranteed.

Quantum Annealing (QA) [3, 4] was introduced as a practical alternative to ASP. QA modifies the ASP framework by incorporating transverse fields to exploit quantum tunneling during the evolution, thereby relaxing the strict adiabaticity requirement.

Both ASP and QA can be formulated within a time-dependent Hamiltonian framework. Let $H_0(t)$ denote the time-dependent Hamiltonian governing the quantum system. Suppose that at the initial time $t = t_i = 0$, the system is in the ground state $|\psi_i\rangle$ of $H_0(t_i)$. Similarly, define the target state $|\psi_t\rangle$ as the ground state of the instantaneous Hamiltonian at $t = t_f > t_i$, with the total evolution duration given by $\tau \equiv t_f - t_i$.

While this framework is broadly applicable to various optimization problems [5], it is constrained by fundamental dynamical limitations. If the evolution proceeds too rapidly, the system will transition into excited states. This limitation is particularly significant in ASP, where the assumption is that the system remains in the instantaneous ground state of $H_0(t)$ throughout the evolution, ensuring that the final state has high fidelity with the target state. This assumption is justified by the adiabatic theorem, which specifies the necessary conditions for successful state preparation.

The primary conditions for adiabaticity are:

1. The evolution must be infinitely slow ($\tau \to \infty$).

2. A nonzero energy gap between the ground and excited states must be maintained throughout the evolution.

A direct consequence of these conditions is that ASP (and to some extent, QA) requires long evolution times to achieve high fidelities. Moreover, if the energy gap closes, the adiabatic assumption fails, leading to transitions into excited states. This issue is particularly relevant in systems exhibiting gap closures due to phase transitions.

In the context of quantum state preparation and annealing, the fidelity or energy of the prepared state serves as a measure of protocol accuracy. In the latter, we shall use the energy as it is easier to optimize. The following sections discuss how to reduce the annealing time of such protocols by optimizing the pulse. In essence, we are making use of methods inspired from quantum optimal control (QOC) [6, 7] and counterdiabatic driving (CD) [8, 9] to increase the gap and allow for faster schedules.

2. *Definition of the instance and Mapping to Qubo*

We start by initializing an instance of the problem. This means choosing the costs c_t, δ_t, the charging power P_n^0, the target energies $\tilde{E}_n$ and the penalty coefficient λ. For a given set of parameters, the QUBO matrix describing the problem is obtained using a Python script. From this matrix, we then obtain the atoms configuration such that they represent the correct Hamiltonian. In order to ensure that the atoms satisfy the hardware constraints, we implement a check on the atoms position to ensure that none of them is over the maximum allowed distance from the origin.

3. *Counterdiabatic Driving (CD)*

Various protocol designs, which are called *shortcuts to adiabaticity*, have been proposed to mitigate the limitations prompted by the adiabatic theorems. The general idea is to shorten as much as possible the evolution time and avoid gap closures in the spectrum. Among the shortcuts-to-adiabaticity methods, Counterdiabatic Driving [8, 10, 11] is a promising candidate, which is formally able to overcome this problem through an ingenious choice of the additional driving term. The basic idea behind CD is to boost adiabatic processes by adding a CD Hamiltonian (the *adiabatic gauge potential*) that suppresses instantaneously transitions between the system eigenstates. In adiabatic schedules with a time-dependent σ_j^z (or equivalently n_j) term, rapid changes can induce diabatic excitations. Adding a CD term counteracts these transitions, ensuring the system follows the adiabatic path more efficiently and reducing the required annealing time.

While the exact adiabatic gauge potential requires exact knowledge of the eigenstates to be computed, we instead aim to learn it. In our case, the CD terms correspond to the $\delta(t)\sigma_i^z$ term in the Hamiltonian. Usually, one optimizes the gauge potential analytically in order to find the proper coefficients $\delta(t)$ that suppress the transitions. In our case, we make no analytical calculation and find the best CD term through a machine learning procedure.
We parametrize the schedules $\Omega(t)$ and $\delta(t)$ by interpolating between l points $\{x_i\}$ (independent for both schedules) and imposing the following conditions

$$\Omega(t) \geq 0 \text{ and } \Omega(t_i) = \Omega(t_f) = 0 \tag{11}$$
$$\delta(t_i) < 0 < \delta(t_f). \tag{12}$$

The two sets of points $\{x_i\}$ are then optimized using a genetic algorithm to minimize the energy cost function. The total number of these points l is a hyperparameter chosen a priori that can enlarge or reduce the complexity of the machine learning procedure. Genetic optimization is a heuristic search method inspired by the principles of natural selection and evolution, widely used for solving high-dimensional and non-convex optimization problems. A genetic algorithm operates by evolving a population of candidate solutions through iterative application of selection, crossover, and mutation. Each individual in the population represents a potential solution, typically encoded as a binary string, real-valued vector. Selection mechanisms favor individuals with higher fitness, ensuring that better solutions contribute more significantly to the next generation. Crossover then recombines genetic material from selected individuals, facilitating exploration of the search space, while mutation introduces small random perturbations to maintain diversity and prevent premature convergence. The process repeats until convergence criteria, such as a maximum number of generations, are met. Genetic algorithms are closely related to machine learning as they aim to solve non-convex complex optimization problems. We compare our optimization to a baseline consisting of a smooth pulse:

$$\Omega_{\text{baseline}}(t)als = \omega_{\max} \sin^2\left(\frac{\pi}{2}\sin\left(\frac{\pi t}{\tau}\right)\right) \tag{13}$$

$$\delta_{\text{baseline}}(t) = -\delta_{\max}\cos\left(\frac{\pi t}{\tau}\right) \tag{14}$$

4. *Choice of annealing time*

After we choose the parametrization of the adiabatic schedule, we need to understand what is the minimum annealing time τ. The adiabatic theorem sets a maximum annealing time which guarantees convergence. Its computation requires knowledge of the energy gap along the drive. However, we can find paths that avoids gap closure using optimized CD schedules. We sweep over different total annealing times τ and run the schedule optimization for each one of them, with the aim of reducing the total annealing time.

II. RESULTS

We show the results obtained for a system composed of $3 \times 3 = 9$ qubits, which represents the process of charging 3 EVs in 3 time steps.

A. Problem instance

The problem instance is determined by the following parameters:

- $P_n^0 = (1, 1, 1)$
- $\delta_t = 0.3c_t$
- $\Delta t = 1$
- $\lambda = 5$
- $P = 1$
- $E_n = (1, 1, 1)$

For the specific instance taken into consideration, the atom configuration is the one shown in Fig 1.

B. Optimal choice of annealing time and schedule

We first perform a sweep over annealing times in order to determine the optimal duration. To do so, we parametrize the drive and the CD term as described in I B 3 and vary the annealing time while simultaneously optimizing the schedule, as explained in previous sections. Given E as the expectation value of the cost Hamiltonian on the final state and E_{gs} the ground state energy, the optimal annealing time is determined from the plot of the cost function versus annealing time, shown in Fig. 2. This corresponds to the shortest time for which the approximation ratio $\left(r = 1 - |\frac{E - E_{gs}}{E_{gs}}|\right)$ reaches a value greater than a threshold r_{min} chosen a priori (in our case con consider $r_{min} = 0.8$). The optimal schedule obtained after optimization for the optimal annealing time is the one shown in Fig 3, along with an example of a standard schedule for comparison.

C. Noise study

After determining the optimal adiabatic schedule, we examine how it schedule behaves under noise. To achieve this, we simulate the execution of the optimal schedule with an added noise channel. We focus on depolarizing noise and tune the strength. Figure 4 illustrates how the approximation ratio evolves as we increase the strength of the

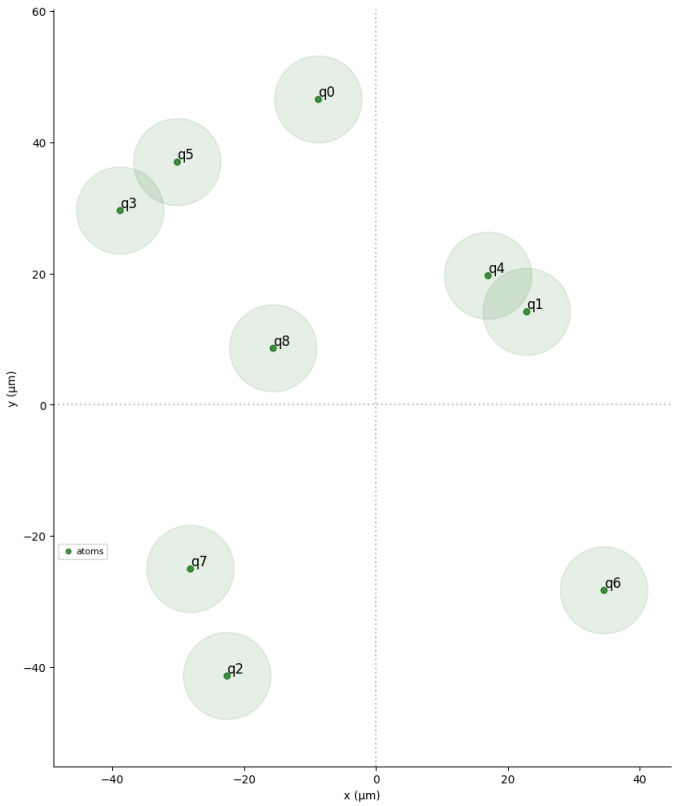


Figure 1: Atoms configuration corresponding to the QUBO matrix for the chosen instance. In top left, we show for comparison the minimum distance from two atoms of the system.

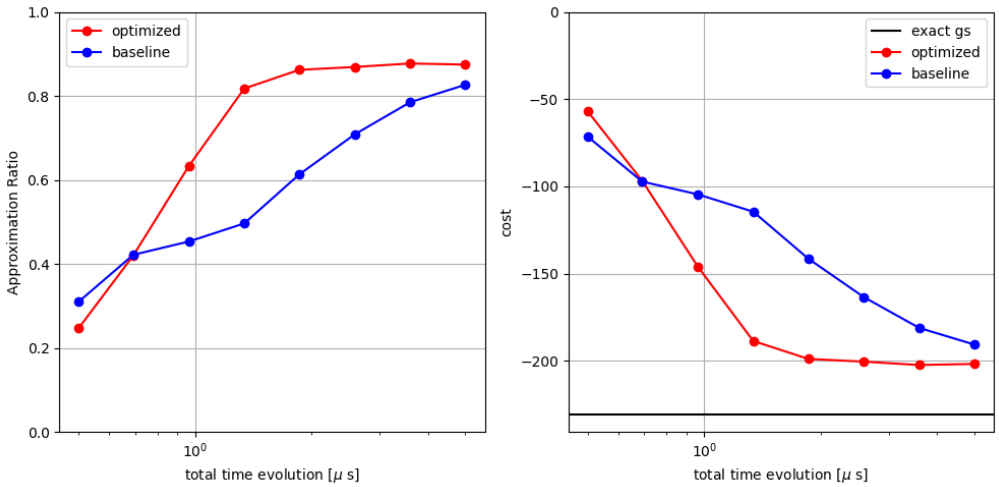


Figure 2: Left: Average cost function evaluated on the final state as a function of the total annealing time τ. Right: Approximation ratio as a function of the total annealing time τ

depolarizing noise. As demonstrated, we observe that the optimal schedule results are barely affected by the noise, as the variation in approximation ratio is always contained within 5% and the final energy is always close to the one obtained with the noiseless simulation.

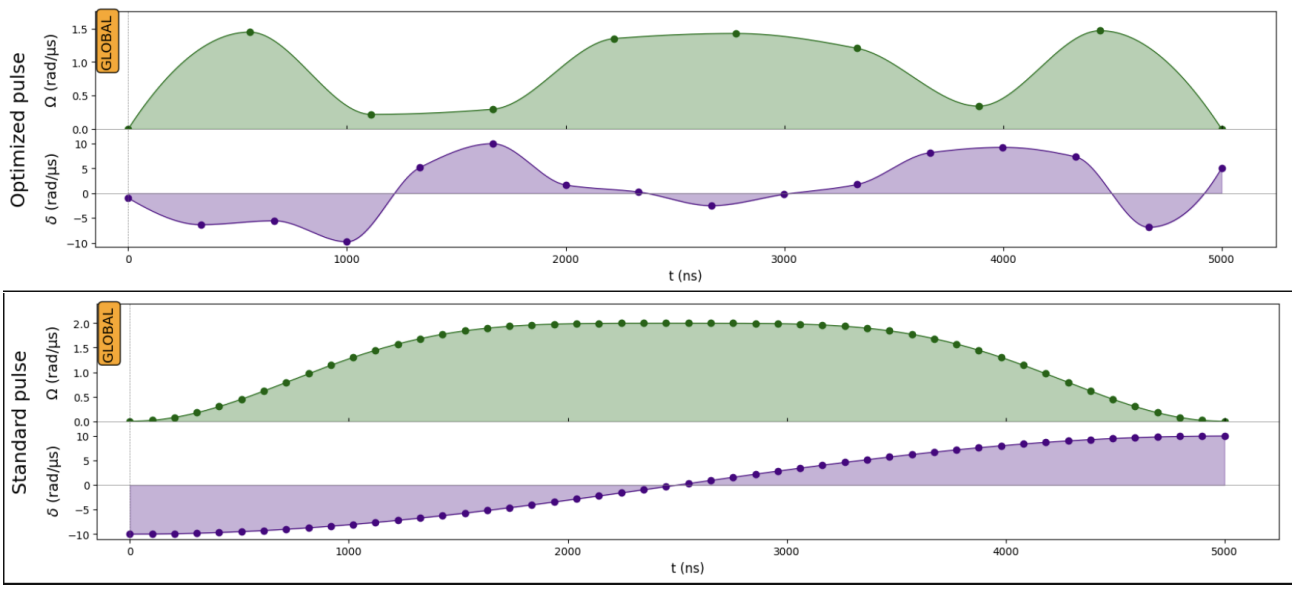


Figure 3: Top: Optimized schedule for drive and CD term for a system of 9 qubits. Bottom: Standard, unoptimized schedule for the adiabatic algorithm also for 16 qubits.

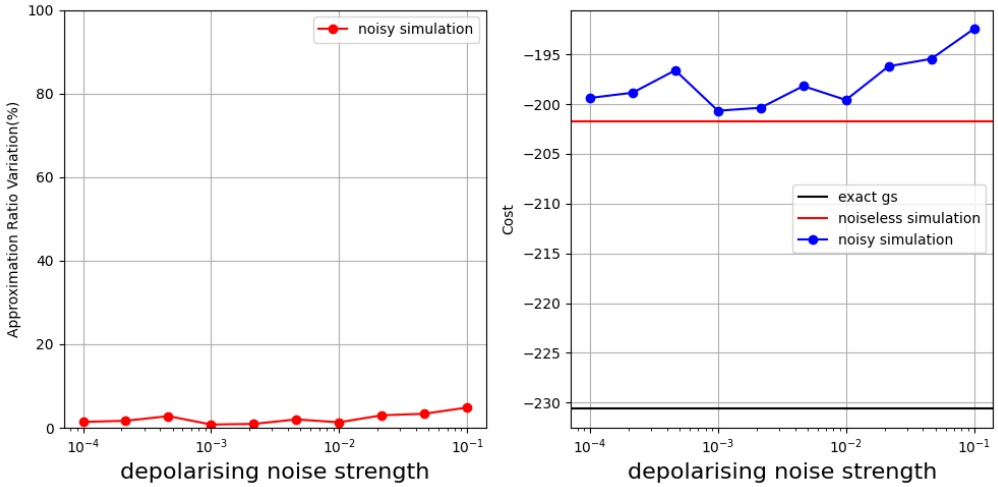


Figure 4: Left: variation in approximation ratio for the 3×3 instance in consideration under the action of depolarizing noise. Right: Final costs obtained for the noisy simulations, compared to the ground state and the result obtained by the noiseless simulation.

D. Study of a 4×4 instance

We performed the same noiseless study also on an instance with $N = 4$ and $T = 4$, resulting in a total of 16 qubits. The problem instance is specified by the parameters:

- $P_n^0 = (1, 1, 1, 1)$
- $\delta_t = 0.3c_t$
- $\Delta t = 1$
- $\lambda = 5$
- $P = 2$

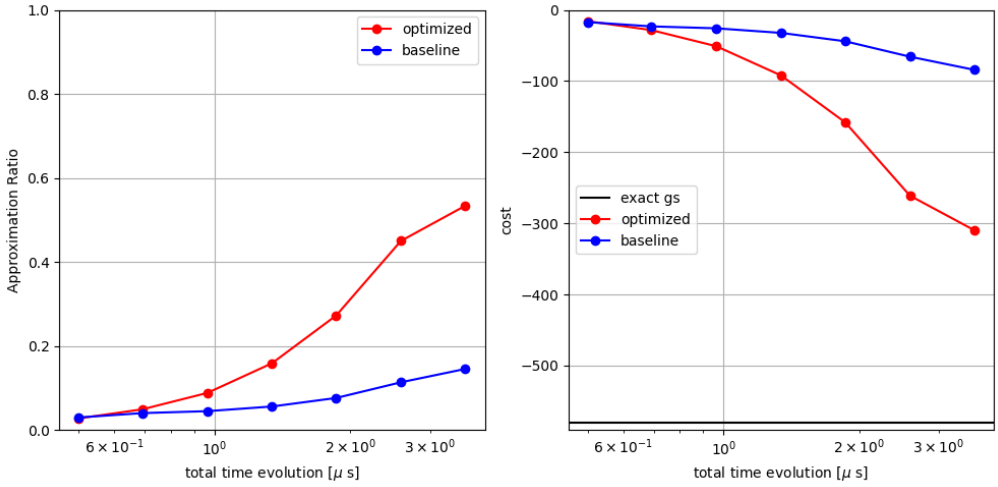


Figure 5: Left: Approximation ratio as a function of annealing time for a 4×4 system. Right: Evolution of the expectation value fo the energy on the final state as a fucntion of annealing time for the same system.

- $E_n = (2, 2, 2, 2)$

The results for approximation ratio and energy are shown in Fig 5

III. FEASIBILITY AND IMPACT

A. Feasibility

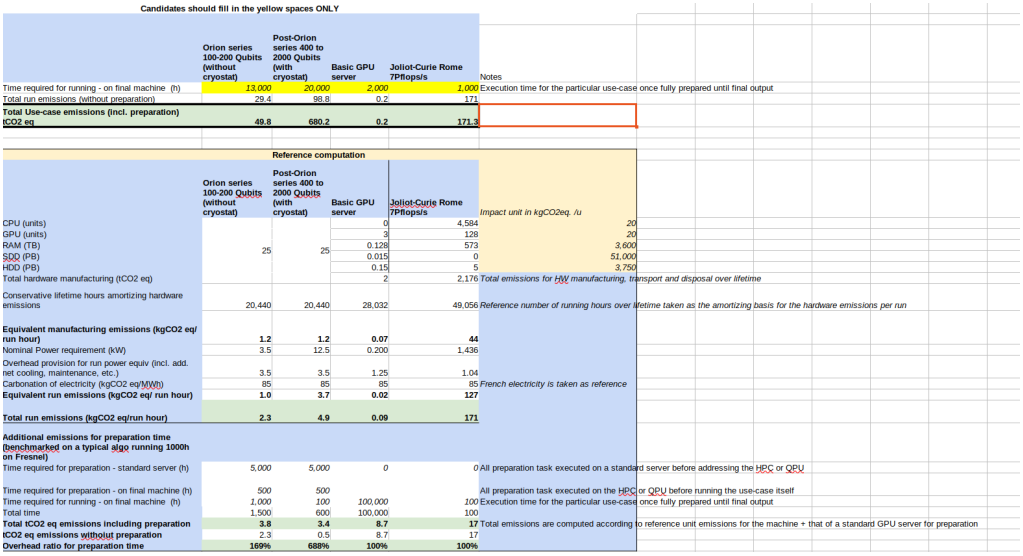
As constraints add complexity, transforming the problem into a QUBO formalism makes it feasible for Pasqal's adiabatic annealing schedule. The training of the schedule scales with the number of interpolating parameters and not with the system size, making it implementable without extreme computational resources even for larger instances. This enables problem-solving without computational clusters, appealing to medium to large companies optimizing vehicle charging schedules. The problem becomes classically hard with 100 to 1000 EVs, yet some instances are currently feasible on Pasqal's hardware, with industry-relevant scales soon within reach.

B. Innovation

Our approach introduces several key innovations that enhance the efficiency and feasibility of solving the bi-directional EV charging problem using Pasqal's quantum computing technology. First, we apply machine learning techniques to optimize the adiabatic schedule, taking inspiration from counterdiabatic driving, instead of relying on predefined annealing schedules. This allows the quantum algorithm to find the most efficient path to the solution, reducing errors and improving accuracy. CD helps prevent unwanted transitions in the quantum system, making the computation faster and more reliable. Secondly, we reformulate the problem so that all cost and penalty terms are integrated into a single quantum optimization framework. This simplifies the problem structure and allows for more efficient execution on quantum hardware. These innovations make our approach more adaptable and scalable, opening up new possibilities for quantum computing applications in energy optimization and other fields.

C. Quantum energy Advantage

Energy benchmark is provided in Fig.6. For our instances, the total QPU time required is given by $T_{\text{QPU}} = N_\tau N_{\text{iter}} N_{\textbf{shots}}\, \mu s = 10 \times 10^3 \times 10^3\, \mu s = 10 \times 10^3 \times 10^3 \mu s$, where N_τ represents the number of adiabatic evolution times tested, N_{iter} is the number of iterations for the optimizer and $N_{\textbf{shots}}$ is the number of shots used to reconstruct the state. The numbers in the spreadsheet are derived under the assumption that for large-scale problems of size $N \times T = O(10^3)$ we must increase both the number of samples and the annealing time by a factor of 10. Additionally,



Candidates should fill in the yellow spaces ONLY

	Orion series 100-200 Qubits (without cryostat)	Post-Orion series 400 to 2000 Qubits (with cryostat)	Basic GPU server	Joliot-Curie Rome 7Pflops/s	Notes
Time required for running - on final machine (h)	13,000	20,000	2,000	1,000	Execution time for the particular use-case once fully prepared until final output
Total run emissions (without preparation)	29.4	98.8	0.2	171	
Total Use-case emissions (incl. preparation) tCO2 eq	**49.8**	**680.2**	**0.2**	**171.3**	

Reference computation

	Orion series 100-200 Qubits (without cryostat)	Post-Orion series 400 to 2000 Qubits (with cryostat)	Basic GPU server	Joliot-Curie Rome 7Pflops/s	*Impact unit in kgCO2eq. /u*
CPU (units)			0	4,584	*20*
GPU (units)			3	128	*20*
RAM (TB)			0.128	573	*3,600*
SDD (PB)	25	25	0.015	0	*51,000*
HDD (PB)			0.15	5	*3,750*
Total hardware manufacturing (tCO2 eq)			2	2,176	*Total emissions for HW manufacturing, transport and disposal over lifetime*
Conservative lifetime hours amortizing hardware emissions	20,440	20,440	28,032	49,056	*Reference number of running hours over lifetime taken as the amortizing basis for the hardware emissions per run*
Equivalent manufacturing emissions (kgCO2 eq/ run hour)	**1.2**	**1.2**	**0.07**	**44**	
Nominal Power requirement (kW)	3.5	12.5	0.200	1,436	
Overhead provision for run power equiv (incl. add. net cooling, maintenance, etc.)	3.5	3.5	1.25	1.04	
Carbonation of electricity (kgCO2 eq/MWh)	85	85	85	85	*French electricity is taken as reference*
Equivalent run emissions (kgCO2 eq/ run hour)	**1.0**	**3.7**	**0.02**	**127**	
Total run emissions (kgCO2 eq/run hour)	**2.3**	**4.9**	**0.09**	**171**	
Additional emissions for preparation time (benchmarked on a typical algo running 1000h on Fresnel)					
Time required for preparation - standard server (h)	*5,000*	*5,000*	*0*	*0*	All preparation task executed on a standard server before addressing the HPC or QPU
Time required for preparation - on final machine (h)	*500*	*500*			All preparation task executed on the HPC or QPU before running the use-case itself
Time required for running - on final machine (h)	*1,000*	*100*	*100,000*	*100*	Execution time for the particular use-case once fully prepared until final output
Total time	1,500	600	100,000	100	
Total tCO2 eq emissions including preparation	**3.8**	**3.4**	**8.7**	**17**	Total emissions are computed according to reference unit emissions for the machine + that of a standard GPU server for preparation
tCO2 eq emissions without preparation	2.3	0.5	8.7	17	
Overhead ratio for preparation time	**169%**	**688%**	**100%**	**100%**	

Figure 6: Energy benchmark for the procedure

a further factor of 10 is introduced to account for the increased number of parameters required for the schedule in larger instances.

[1] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution (2000), arXiv:quant-ph/0001106 [quant-ph].

[2] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem, Science **292**, 472 (2001), https://www.science.org/doi/pdf/10.1126/science.1057726.

[3] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse ising model, Phys. Rev. E **58**, 5355 (1998).

[4] T. Albash and D. A. Lidar, Adiabatic quantum computation, Reviews of Modern Physics **90**, 10.1103/revmodphys.90.015002 (2018).

[5] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, Quantum annealing for industry applications: Introduction and review, Reports on Progress in Physics (2022).

[6] S. J. Glaser, U. Boscain, T. Calarco, C. P. Koch, W. Köckenberger, R. Kosloff, I. Kuprov, B. Luy, S. Schirmer, T. Schulte-Herbrüggen, *et al.*, Training schrödinger's cat: Quantum optimal control: Strategic report on current status, visions and goals for research in europe, The European Physical Journal D **69**, 1 (2015).

[7] I. Khait, J. Carrasquilla, and D. Segal, Optimal control of quantum thermal machines using machine learning, Phys. Rev. Res. **4**, L012029 (2022).

[8] M. V. Berry, Transitionless quantum driving, Journal of Physics A: Mathematical and Theoretical **42**, 365303 (2009).

[9] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement learning in different phases of quantum control, Phys. Rev. X **8**, 031086 (2018).

[10] M. Demirplak and S. A. Rice, Adiabatic population transfer with control fields, The Journal of Physical Chemistry A **107**, 9937 (2003), https://doi.org/10.1021/jp030708a.

[11] M. Demirplak and S. A. Rice, Assisted adiabatic passage revisited, The Journal of Physical Chemistry B **109**, 6838 (2005).