

Using quantum annealing to generate test cases for cyber-physical systems

HUGO ARAUJO, King's College London, United Kingdom

XINYI WANG, Simula Research Laboratory, Norway

MOHAMMAD REZA MOUSAVI, King's College London, United Kingdom

SHAUKAT ALI, Simula Research Laboratory, Norway

Quantum computing offers the potential to address complex computational challenges, particularly in areas such as simulation and optimisation. Despite this promise, current gate-based quantum devices remain limited in scale and reliability, constraining their practical use. In this context, quantum annealing (QA) emerges as a viable and currently accessible alternative for solving optimisation problems, as it employs adiabatic quantum evolution to efficiently explore large and complex search spaces. This makes it a natural candidate for search-based software testing in the Cyber-Physical Systems (CPS) domain, which demands effective test cases due to their complexity and combination of continuous and discrete domains. This work explores the use of quantum annealing to enhance test case generation for CPS through a mutation-based approach. We employ quantum annealing to identify areas on test cases that can be improved and apply mutations to those areas. We encode test case mutation as a binary optimisation problem, and use quantum annealing to identify and target critical regions of the test cases for improvement. Our approach mechanises this process into an algorithm that uses D-Wave's quantum annealer to find the solution. As a main contribution, we offer insights into how quantum annealing can advance testing methodologies by empirically evaluating the correlation between problem size, hardware limitations, and the effectiveness of the results. Moreover, we compare the proposed method against state-of-the-art classical optimisation algorithms, targeting efficiency (time to generate test cases) and effectiveness (fault detection rates). Our results indicate that quantum annealing may enable faster test case generation (when problem decomposition, sampling and scheduling strategies are applied) while achieving comparable fault detection performance to state-of-the-art alternatives.

CCS Concepts: • **Computer systems organization** → **Embedded and cyber-physical systems**.

Additional Key Words and Phrases: quantum computing, quantum annealing, test case generation, CPS

ACM Reference Format:

Hugo Araujo, Xinyi Wang, Mohammad Reza Mousavi, and Shaukat Ali. 2026. Using quantum annealing to generate test cases for cyber-physical systems. 1, 1 (September 2026), 33 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Quantum computing integrates quantum mechanics and computer science to change how complex computational problems can be analysed and solved. It takes advantage of quantum mechanics principles such as superposition and entanglement to analyse several possible solutions simultaneously, a phenomenon often referred to as *quantum parallelism* [59]. *Superposition* means that qubits, the quantum counterpart to classical bits, can represent a combination

Authors' addresses: Hugo Araujo, hugo.araujo@kcl.ac.uk, King's College London, United Kingdom; Xinyi Wang, xinyi@simula.no, Simula Research Laboratory, Norway; Mohammad Reza Mousavi, mohammad.mousavi@kcl.ac.uk, King's College London, United Kingdom; Shaukat Ali, shaukat@simula.no, Simula Research Laboratory, Norway.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

of values (typically represented as computational bases $|0\rangle$ and $|1\rangle$) simultaneously. *Entanglement* further allows correlations between qubits that cannot be reproduced classically, enabling coordinated state changes across multiple qubits when acted upon by quantum gates.

The advantages of quantum parallelism does not stem from simply running many computations in parallel in the classical sense, but from the mathematical structure of superposition and entanglement. A quantum set of n qubits in superposition encodes 2^n computational states simultaneously, which allows quantum algorithms to manipulate entire probability amplitude distributions with a single unitary transformation. The key benefit arises from the ability of quantum algorithms to design constructive and destructive interferences so that probability amplitude is reinforced on desirable solutions and suppressed on undesirable ones [33, 54, 59]. These principles are utilised in some of the earliest demonstrations of theoretical quantum advantage. Shor’s algorithm achieves an exponential speedup for integer factorisation by exploiting interference over periodicity [70], while Grover’s search algorithm provides a quadratic speedup for unstructured search problems by amplifying the amplitude of certain states [36].

Therefore, classical algorithms must explore the search space explicitly or rely on heuristics to approximate structure, whereas quantum algorithms can globally act on large portions of the solution space through a small number of operations. For optimisation problems, quantum methods such as adiabatic quantum computation [2] make use of this structure to guide the system toward solutions more efficiently under certain problem encodings. While quantum parallelism does not guarantee a speedup for arbitrary problems, it enables algorithmic behaviour via global updates that cannot be replicated by classical randomised or local-search heuristics.

In particular, quantum annealing [64], one of the quantum algorithms that exploits these properties, stands out as a promising method for solving optimisation problems. More specifically, quantum annealing is an optimisation algorithm based on adiabatic quantum computation, in which a quantum system comprising many qubits is gradually evolved from an initial superposition of all possible states toward a final state representing the solution.

This algorithm is implemented on specialised hardware devices known as *quantum annealers*, such as those developed by D-Wave [24]. Unlike gate-based devices, which currently operate with far fewer qubits, contemporary D-Wave systems can utilise over 5000 qubits [25], enabling substantially larger problem instances. These devices operate on problems encoded in a Quadratic Unconstrained Binary Optimisation (QUBO) formulation, a mathematical representation that expresses an optimisation problem as a quadratic objective function over binary variables. Although D-Wave’s annealers use QUBO as their native input format, quantum annealing as an algorithm is more general and can be defined independently of any specific hardware architecture.

Quantum algorithms, including quantum annealing, have the potential to offer performance benefits for certain classes of optimisation problems by exploiting quantum parallelism properties [54]. However, demonstrating consistent quantum advantage in practice remains challenging: contemporary quantum devices are still subject to noise, limited qubit connectivity, and architectural constraints, even though isolated demonstrations of advantage exist for specific tasks on gate-based hardware [8]. Consequently, effectively making use of quantum computing for optimisation requires addressing several practical challenges, such as formulating problems in hardware-compatible representations and embedding them efficiently onto the underlying qubit architecture. In particular, (i) encoding the optimisation task into a QUBO, and (ii) minimising the qubit and coupler resources required for embedding, remain key considerations for achieving reliable performance on current quantum annealers.

As an application domain that can take advantage of quantum computing, search-based software testing [47, 48, 62] is a prime candidate, as it involves optimising search techniques to generate test cases. This is particularly true in the subject domain of Cyber-Physical Systems (CPS), due to their intricate nature and large search space. CPS are

integrations of computational, network, and physical components that interact with each other and form the backbone of modern applications such as autonomous vehicles, industrial automation, and smart grids [68]. Their safety-critical nature and continuous interaction with dynamic environments make testing and optimisation essential to ensure reliability and performance [44]. Essentially, effective CPS testing requires a robust test case generation strategy and one of its primary challenges lies in generating test suites that are both diverse and effective at detecting faults, without incurring prohibitive computational costs. Typical approaches either use efficient randomised methods, such as fuzzing [78], or effective search-based methods, via, for instance, optimisation solving [6]. Despite their utility, these strategies often present limitations in balancing testing effectiveness with computational efficiency due to the large problem search space.

As the main objective of this work, we aim to explore the use of quantum computing to efficiently generate test cases for CPS via a mutation-based, smart fuzzing approach. Mutation in the context of test case generation involves introducing changes or “mutations” into an existing set of test cases to create new ones. Instead of traditional, purely random mutations [71], we employ a smart approach to identify when and how they are applied. Our approach uses quantum annealing to determine suitable regions within each test case. Here, a test case is represented as a trajectory of system variables over time, and the suitable regions correspond to the specific temporal intervals where mutations are introduced to maximise behavioural divergence or fault-detection potential. To determine these regions, we use metrics of testing adequacy (such as measures of effectiveness) to populate the objective formula. Then, we generate and embed mutations into the test cases to improve their effectiveness in terms of failure detection and improve their coverage in terms of diversity. We mechanise our approach into an algorithm that encodes the problem into a binary optimisation one and makes calls to the D-Wave’s quantum annealer to solve it.

We show the effectiveness of our approach (measured in terms of the fault detection rate) and its efficiency (measured as the time required to generate a complete set of test cases, including QUBO construction, embedding, and annealing execution) by empirically evaluating the resulting test suite against those produced by alternative classical optimisation algorithms. The dataset, code, and models used in our experiments are publicly available¹. The results collectively indicate that quantum annealing may be able to generate test cases more efficiently than the classical alternatives (when robust problem decomposition, sampling, and scheduling strategies are involved) while achieving comparable fault detection rates.

While the underlying methodology of our approach to mutation-based test case generation could, in principle, be applied to other software domains, our motivation, design choices, and formulation are closely tied to the unique challenges and inner workings of cyber-physical systems (CPS). In particular, our solution is deeply integrated with the semantic model of CPS, especially the representation of system trajectories that capture the evolution of continuous variables over time. This dependency on trajectory-based semantics is fundamental to how mutations and divergences are defined and evaluated. Not all safety-critical systems share this continuous or hybrid nature, which makes the proposed approach particularly relevant to CPS rather than purely discrete or event-driven systems.

1.1 Research Questions

The approach presented in this work aims to study the efficiency and effectiveness of test generation for cyber-physical systems (CPS) using quantum computing principles. The intended impact of our results is twofold: first, to demonstrate that quantum annealing (QA) can serve as a practical and competitive optimisation technique for guiding CPS test

¹The repository can be found at: <https://doi.org/10.5281/zenodo.17965074>

mutations, and second, to provide quantitative evidence that QA achieves comparable fault-detection capability and computational effort relative to classical baselines. By showing that QA can be effectively applied to real CPS models, this work contributes towards establishing quantum optimisation as a viable tool in testing pipelines.

The approach presented in this work is based on optimising CPS tests by applying mutations to inputs. In our context, such inputs are typically given as trajectories representing variable's valuations over time. Given the two-dimensional space (temporal and spatial) of the inputs, the choice of when and how to modify them has a crucial impact on the efficiency and effectiveness of the test mutation strategy.

As a response to these challenges, this work embeds a quantum algorithm into a systematic process that identifies and mutates key areas of CPS test inputs. We hypothesise that applying quantum annealing to carefully select the regions of inputs to mutate has a significant impact on test case effectiveness. Additionally, we hypothesise that splitting the problem into sub-problems and using quantum annealing results in improved effectiveness compared to only using a classic state-of-the-art alternative. We turn our hypotheses into the following research questions.

- **RQ1:** When using quantum annealing, what is the correlation between problem size, the number of physical qubits, and effectiveness of the results?
- **RQ2:** Is using quantum annealing to mutate the test cases more effective compared to the alternatives?
- **RQ3:** Is using quantum annealing to mutate the test cases more efficient compared to the alternatives, based on experimental results and theoretical analysis?

To answer these research questions, we evaluate our hypotheses via a controlled experiment that makes use of two CPS case studies (a platoon of connected vehicles and a pneumatic suspension system). These systems represent two distinct types of safety-critical CPS: the former operates over a layer of communication networks, while the latter is strictly offline. The first research question relates to optimising the number of qubits and their correlation with the effectiveness of the results. The optimal values found via RQ1 are used to answer the remaining research questions. The second and third research questions relate to the use of quantum annealing and how it fares against the baselines.

1.2 Paper Structure

The remainder of this paper is organised as follows. In Section 2, we discuss the related work. In Section 3, we describe the preliminary notations and techniques used in our work. In Section 4, we present our process on how to use quantum annealing to identify critical areas of a previously generated test suite and how to improve them by embedding mutations. In Section 5, we describe the design, execution, and results of our controlled experiment that aims to answer our research questions. Finally, in Section 6, we draw some conclusions and point out the directions of our future research.

2 RELATED WORK

Quantum computing research is rapidly advancing in various fields, such as finance [32], drug discovery [15], and chemistry [56]. The application of quantum computing in software engineering is also emerging in recent years, especially in software testing [74–76]. Industrial applications illustrate this trend, such as Bosch [11] and BMW [11], who have reportedly used quantum computing for software testing. In addition, the elevator company Orona showcases a successful application of quantum extreme learning machines for regression testing [74]. In research fields, Miranskyy applies Grover search to enhance the efficiency of dynamic testing on classical programs [53].

More closely related to our work, quantum computing approaches are being investigated to be used in the context of test case optimisation. For instance, Hussein et al. [38] uses amplitude amplification to solve test case minimisation problems via two quantum search algorithms. Wang et al. [75] used a QAOA-based algorithm (which aims to minimise a cost function via finding the lowest values for the energy function of a quantum system) to solve test case optimisation problems. Moreover, BootQA [76] is a tool used to solve test case minimisation problems with quantum annealing. The tool employs a sampling technique to select test cases from a test suite, and feeds them to a quantum annealer through QUBO formulas. Unlike our work, none of the approaches mentioned above focus specifically on test case generation. More precisely, we employ quantum computing to identify the critical regions of test cases to be improved and, to our knowledge, no existing work in the literature employs a quantum algorithm in the generation of test cases.

In this work, we employ quantum annealing rather than the above-mentioned alternative algorithms, such as QAOA, because QAOA and other variational algorithms operate on gate-based quantum computers, such as IBM's superconducting quantum devices. These systems are currently constrained by limited qubit counts (for instance, one of the most powerful publicly available systems, *ibm_pittsburgh*, contains 156 qubits) and restricted qubit connectivity. In addition, our approach depends on a fully connected QUBO formulation, which leads to extremely deep transpiled circuits. Current quantum computers, however, can only handle relatively shallow circuits [45]. Currently, as far as we are aware, the deepest QAOA implementations have circuit depths of approximately 150 [19, 60]. Extremely large depths cause the computation to completely decohere, rendering the optimisation ineffective [63].

To that end, however, many classical model-based test case generation approaches for CPS can be found in the literature [4, 6, 29, 31, 42, 51, 80]. Sadri-Moshkenani et al. [69] published a comprehensive survey on state-of-the-art techniques used in this context. Several of these approaches employ a search-based heuristic to optimise a fitness function and guide the search to effective test cases, such as Bayesian optimisation [31], genetic algorithm [4], and NSGA-II [6]. Arrieta et al. [7] utilises mutation operators and NSGA-II to mutate test cases and generate additional ones while considering several criteria such as test case similarity and test execution time. Our work, instead, specifically focuses on exploring the use of quantum annealing as an alternative to the classical heuristics used in many of the examples above. As a contribution to this work, in our controlled experiment, we provide empirical evidence that quantum annealing can be considered as a suitable alternative the classical counterparts.

Within the broader context of test case generation for a general class of systems, mutation-based fuzzing is a technique that aims to generate new test cases by mutating original ones, either randomly or with pre-defined mutation strategies [20–22]. In certain contexts, steps can be taken to prevent test cases being rejected due to being invalid input sequences. For instance, one can set up constraints in the inputs space, such as rates of change in variable values or, in the context of code fuzzing, through grammar rules [41, 65]. Optimisation heuristics such as simulated annealing [16] and Markov Chains [17] have also been used in the literature to optimise mutation-based fuzzers. As related work, Nichols et al. proposed a method for castor fuzzing [58] by incorporating Generative Adversarial Network (GAN) models. Böttinger [18] formalised fuzzing as a reinforcement learning problem, where the fuzzing agent generates effective inputs by observing rewards caused by different mutations. We would like to reinforce that our work differs from the ones above by, instead of using classical algorithms, incorporating a quantum optimisation algorithm to mutate the test cases aiming for higher effectiveness and efficiency.

Conversely, there is another body of research on mutation generation and testing on quantum software, which demonstrates growing interest in adapting classical fault-based testing strategies to the quantum domain. For example, mutation generation tools such as QMutPy [34, 35] and Muskit [50] aim to generate mutated quantum programs systematically with mutation operators. MutTG [77] is a mutation-based approach to generate the minimal test suite

that can maximize the number of detected mutations, thereby reducing the testing cost. These studies primarily target gate-based quantum circuits and focus on the definition of mutation operators, efficiency of mutant generation, and fault models at the circuit level. In contrast, in our work leverages QA to identify and modify critical regions within CPS test trajectories, enabling the generation of behaviour-divergent test cases that can support more effective testing in CPS systems.

3 BACKGROUND

In this section, we discuss the preliminary knowledge used throughout this work.

3.1 Cyber-Physical Systems

Cyber-physical systems (CPS) integrate computational systems into their physical environments. They represent a combination of discrete and continuous dynamics, often found in systems where a digital controller unit (discrete) is connected with some physical system (continuous).

Particularly, a CPS might involve complex networks of connected components; i.e., physical, computational, or network components that communicate internally within the same entity, or externally with a separate entity. Together, they coordinate sensing, control, and actuation processes, often also involving human interaction. Examples of modern CPSs include vehicles and robotic systems [55]. Such systems require rigorous verification methodologies since they are usually safety-critical. The concept of a cyber-physical system is a generalisation of embedded systems, a collection of computing devices communicating with each other and interacting with the environment [3]. This integration allows them to exhibit autonomous and adaptive behaviour across domains such as transportation, manufacturing, and energy [37]. As these systems become increasingly common in safety-critical settings, ensuring their correctness under realistic and variable operating conditions demands systematic approaches to testing and optimisation.

As an example of a complex CPS, consider a platoon of connected vehicles composed of a human-driven leading car followed by one or more autonomous vehicles. The system integrates computational controllers within each car, a communication layer that applies vehicle-to-vehicle protocols, and control elements that interface with the vehicle kinematics and dynamics (such as acceleration, braking, and gear switching). The human driver provides the initial input to the system, while the autonomous vehicles continuously adjust their behaviour based on sensor data and communication feedback. Over time, this interaction allows the vehicles to maintain a safe distance between them. This system, referred to as a platoon of connected vehicles, is presented as one of our case studies in Section 5.2. The second case study analyses an automotive pneumatic suspension system, which enhances driving comfort by maintaining the chassis level and compensating for road irregularities. The system consists of valves installed at each wheel, a central compressor, and an escape valve that jointly regulate the air pressure to adjust the vehicle height.

Many classes or representations of cyber-physical systems can be found in the literature [3, 27]. In this work, we make use of Matlab/Simulink [30] - a widely-used development and simulation language for CPS - to implement models of CPS. The semantics of these models is represented by variable valuations and trajectories, formalised below:

DEFINITION 1 (VALUATION). *Given a set of variables $V = \{X_1, \dots, X_n\}$, we denote by $Val(V) = V \rightarrow \mathbb{R}$ the set of all total functions from V to the real domain $\mathbb{R}$.*

The notion of trajectory maps the moments in time to variable values.

DEFINITION 2 (TRAJECTORY). *Given a set of variables V , the set of trajectories over V , denoted by $Trajs(V) = \{x_1, \dots, x_m\}$, is the set of all mappings $T \rightarrow Val(V)$, where T is the time domain, henceforth assumed to be a convex subset of $\mathbb{R}_{\geq 0}$.*

Essentially, trajectories describe a system's behaviour over time. In this work, both input and output variables are represented by trajectories. Typically, when testing CPS, a trajectory that comprises the set of input variables forms a test case and its verdict is given by performing checks on the corresponding output trajectory.

3.2 Quantum Computing

Quantum computing is a computational paradigm that applies principles of quantum mechanics to perform computation. Unlike classical computing, which use bits, quantum computing uses quantum bits (*qubits*). There are two primary paradigms in quantum computing: *gate-based quantum computing* and *adiabatic quantum computing* (AQC). The former performs calculations through sequences of quantum gates on different qubits in discrete time steps, where each gate corresponds to a specific Hamiltonian² for a finite duration. The latter relies on a continuous and gradual evolution from a simple Hamiltonian to a final Hamiltonian whose ground state encodes the solution to the corresponding problem. Quantum annealing is a practical realization of this adiabatic paradigm.

3.2.1 Theoretical Foundations. Since QA realizes adiabatic quantum computing for optimisation tasks, its underlying mechanism is rooted in the *adiabatic theorem*, which states that if a quantum system is in the ground state of a Hamiltonian, and the Hamiltonian evolves 'sufficiently slowly', the system will remain in the ground state throughout the evolution and end in the ground state of the final Hamiltonian. In AQC, this theorem can be utilized for computation. Specifically, a quantum system is first prepared in the ground state of an easy-to-solve, initial Hamiltonian, and then the Hamiltonian is gradually transformed into a complex final Hamiltonian whose ground state encodes the solution to the problem to solve. If the quantum system evolves slowly enough, it will remain in the ground state and end up at the ground state of the final Hamiltonian. When the final Hamiltonian is expressed as an Ising Hamiltonian [23], whose ground state corresponds to the solution of an NP-hard optimisation problem, AQC provides the quantum approach to tackling the problem through the adiabatic evolution. The Hamiltonian evolution process $H(t)$ can be expressed as Equation 4.

$$H(t) = A(t)H_{\text{initial}} + B(t)H_{\text{problem}}, \quad (1)$$

where time $t \in [0, T]$, H_{initial} represents the initial Hamiltonian, while H_{problem} denotes the complex Hamiltonian that encodes the problem to be solved. The coefficients $A(t)$ and $B(t)$ control the relative influence of these two Hamiltonians during the annealing process. At the beginning ($t = 0$), the initial Hamiltonian dominates ($A(0) \gg B(0)$). By the end of the gradual evolution ($t = T$), the problem Hamiltonian dominates ($A(T) \ll B(T)$), ideally leaving the system in the ground state of H_{problem} , which corresponds to the optimal solution.

Typically, the initial Hamiltonian is represented by

$$H_{\text{initial}} = \sum_{i \in V} \sigma_i^x, \quad (2)$$

where σ_i^x is the x -Pauli operator acting on the i -th qubit. This Hamiltonian applies equal amplitude superpositions on all qubits to ensure that all possible states are explored simultaneously. V denotes the set of qubits. The final Hamiltonian is given by

$$H_{\text{problem}} = \sum_{i \in V} h_i \sigma_i^z + \sum_{(i,j) \in E} J_{ij} \sigma_i^z \sigma_j^z, \quad (3)$$

where σ_i^z is the z -Pauli operator acting on the i -th qubit. J_{ij} is coupling strength between qubit i and j . h_i is the bias term on qubit i . E is the set of edges couplings in the Ising graph.

²The total energy of a quantum system.

At the end of the evolution ($t = T$), the contribution from the initial Hamiltonian vanishes, and the system behaves as a purely classical Ising model. After the measurement, the qubits collapse to classical spin variables, yielding the classical Ising energy function:

$$H_{\text{Ising}} = \sum_i h_i s_i + \sum_{i,j} J_{ij} s_i s_j, \quad (4)$$

where all Pauli operators are replaced by classical spin variables $s_i \in \{-1, +1\}$.

However, despite the theoretical support of the adiabatic theorem for AQC, maintaining strict adiabaticity in real quantum devices is difficult. The background noise and environment can disturb the system and push it out of the ground state. In addition, evolving the Hamiltonian ‘sufficiently slowly’ is not trivial for physical implementations. Consequently, QA adopts a more flexible, heuristic version of QAC, as the Hamiltonian annealing schedule from H_{initial} to H_{problem} is determined heuristically rather than strictly satisfying adiabatic conditions. As a result, QA does not guarantee that the system remains in the ground state throughout the process, but instead maintains a non-zero probability of reaching the ground-state solutions, making it an effective heuristic optimisation method in practice.

3.2.2 Implementation of Quantum Annealing. Quantum annealing is highly effective for tackling combinatorial optimisation problems, which aim to search through a large set of possible solutions to identify the optimal solution among all possibilities. It takes advantages of quantum properties (e.g., superposition, entanglement, and tunnelling) to explore solution spaces more efficiently than classical methods. Figure 1 illustrates how quantum annealing is executed on a D-Wave quantum annealer, the hardware that realizes this method. It comprises three main steps:

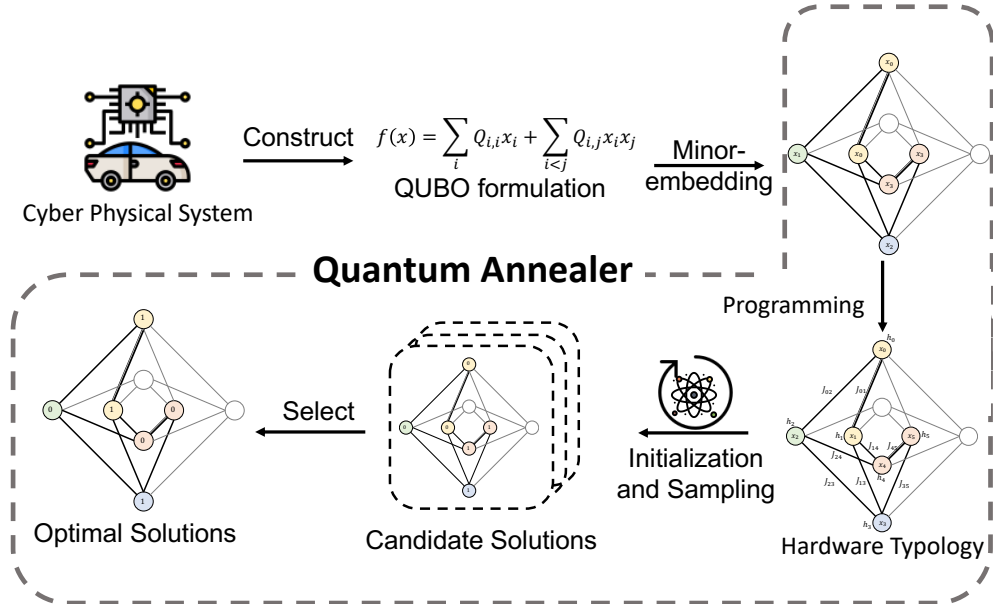


Fig. 1. Quantum Annealing process that runs on D-Wave systems. A QUBO formulation is formed based on the problem, which is mapped onto the hardware via the minor embedding and programming phase. The optimal solutions are filtered out of the set of candidate solutions obtained from multiple annealing runs.

Step 1. QUBO construction: To apply QA to an optimisation task, the problem should first be expressed in a form compatible with the hardware. This is typically achieved using either the Ising model, as explained previously, or the

Quadratic Unconstrained Binary Optimisation (QUBO) [67] formulation. These two representations are mathematically equivalent and can be transformed into one another through a simple linear mapping $s = 2x - 1$ and $x \in \{0, 1\}$. In this work, we adopt the QUBO formulation, as it offers a convenient binary representation for modeling combinatorial optimisation problems. The QUBO model is defined as follows:

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} = \sum_i Q_{i,i} x_i + \sum_{i < j} Q_{i,j} x_i x_j \quad (5)$$

Similar to traditional optimisation algorithms, the QUBO formulation $f(\mathbf{x})$ (i.e., objective function) evaluates each candidate solution $\mathbf{x}$ in terms of fitness values, which describes the energy of a quantum system. Here $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]$ is a vector consisting of binary decision variables x_i , and $\mathbf{x}^T$ represents its transpose. The matrix $\mathbf{Q}$ contains all coefficients in the objective function, whose values are determined by the specific optimisation problems, as defined by domain experts. The rightmost part of Equation 5 represents the expanded equation. Considering the implementation of the quantum system, each decision variable x_i represents a node of the quantum system (i.e., a qubit). Each qubit has a *bias*, which is defined by the linear coefficients $Q_{i,i}$ of a corresponding variable³. Moreover, qubits interact through couplers, where the two corresponding qubits are entangled, and the *coupling strength* is decided by the quadratic coefficients $Q_{i,j}$, ($i < j$) between two corresponding variables. Quantum annealing aims to find the optimal solution that can minimise the fitness value according to the $\mathbf{Q}$ matrix.

QUBO was chosen because it is reportedly the native, and best-supported input format for current D-Wave hardware [39]. Alternatives such as Constrained Quadratic Model (CQM), Constraint Satisfaction Problems (CSP), and Higher-order Unconstrained Binary Optimization (HUBO) indeed offer enhanced expressive power and improve fidelity, but they often come with higher embedding costs or are less mature in terms of hardware support. In addition, all those formulations are ultimately transformed into a QUBO or Ising model representation in the final stage for processing, since QA operates on such problems [25, 26, 40].

Step 2. Minor-embedding and programming: The shape in which the physical qubits are physically or logically connected in hardware is called connection topology. In the process of *programming* a QUBO formulation into the hardware, a QUBO formula is first mapped into a QUBO graph. During the programming step extra qubits may be required since current computers have limited connection topology. Hence, the original connection graph of the QUBO needs to be transformed into a new graph to fit the hardware topology by introducing *qubit chains* in a process called *minor-embedding*.

Consider the example in Figure 2. Figure 2a shows a fully-connected QUBO graph with four variables, while Figure 2b depicts the topology supported by the D-Wave *Advantage* system, where not all qubits are directly connected by couplers. Thus, in some cases, one variable should be mapped to a qubit chain (multiple connected qubits in the topology) to ensure any two pairs of the four variables are connected physically on the hardware. As shown in Figure 2b, x_0 and x_3 are both represented by a qubit chain of two qubits to remain connected to x_1 and x_2 as well as to each other. After the minor-embedding, in the programming step, all the coefficients of Q_i and $Q_{i,j}$ will be assigned to the qubit bias and coupler.

Step 3. Initialisation and sampling: When the optimisation process begins, all qubits are first *initialised* in the ground state of an easy-to-prepare initial Hamiltonian, where all qubits are in an equal superposition of all possible states. Then, the quantum system evolves from the initial Hamiltonian to the final Hamiltonian, which encodes the QUBO formulation of the optimisation problem to be solved, and its ground state encodes the solution. However, since

³Since x_i is a binary variable, it holds that $x_i^2 = x_i$. Therefore, the quadratic term $\sum_i Q_{i,i} x_i^2$ can be expressed as the linear term $\sum_i Q_{i,i} x_i$.



Fig. 2. QUBO formalisation for a four-variable problem. The theoretical QUBO graph (a) and its representation when mapped onto the D-Wave Advantage system hardware (b).

it is difficult for quantum annealing to adhere to the adiabatic theorem, the output is not always guaranteed to be the exact solution. To minimise this uncertainty, multiple initialisation and sampling processes are iteratively performed, generating several *candidate solutions*. The solution with the lowest energy is generally regarded as the *optimal solution*.

3.2.3 Scheduling in Quantum Annealing. As explained in the previous section, the *annealing schedule* specifies how the coefficients $A(t)$ and $B(t)$ vary throughout the total annealing time T . The most common schedule is linear, where the transition between $A(t)$ and $B(t)$ occurs at a constant rate. However, in practical settings, custom or nonlinear schedules can be defined to adapt the rate of change at different points of the anneal.

A particularly useful scheduling technique (possible in D’Wave systems) is the pause-and-quench method. In this approach, the annealing process is intentionally paused at a specific intermediate value of the annealing parameter to allow the system to relax thermally or quantum mechanically toward lower-energy states. This pause can essentially enhance the probability of remaining in the ground state, improving solution quality. Following the pause, the anneal is *quenched*, meaning it is completed by increasing $B(t)$ (and reducing $A(t)$) towards their final values. Moreover, users are allowed to specify the evolution through a sequence of pause-quench pairs, which accelerates or slows down the evolution as required.

4 QUANTUM ANNEALING FOR TEST CASE GENERATION

In this section, we describe our methodology for generating test cases. We start with an overview of the process (Section 4.2), followed by a detailed description of each of its steps: the generation of seed test cases and the collection of metrics (Section 4.3), solving the optimisation problem (Section 4.4), and, lastly, the mutation of test cases (Section 4.5).

4.1 Running Example

Consider as our running example a CPS that describes a minimalistic vehicle engine that receives the position of the accelerator pedal as input (valued between 0 and 1) and outputs the amount of fuel (valued between 0mL and 5mL) to be injected into cylinders. For simplicity, this system is designed to directly map the position of the pedal to the amount of fuel. For instance, when the pedal is on 0, the amount of fuel to be injected should be 0 mL. Conversely, when the pedal is on 1 the amount of fuel released should be 5 mL. Assume that the test engineers of this system have created a simulation test model that faithfully captures the intended behaviour. However, a hardware failure in the physical implementation resulted in the engine releasing more fuel than it should (by 0.2 mL). To test this system, test

engineers generate test cases and compare the output of the simulation model (expected output) against that of the system implementation (observed output), as depicted in Figure 3.

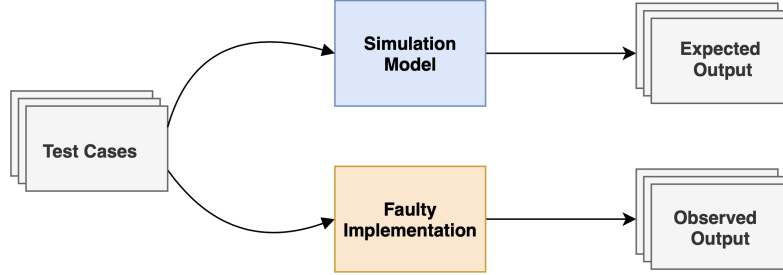


Fig. 3. Process for testing the running example. Test cases are executed both on the system implementation and on a simulation model and the outputs are compared for divergence.

In what follows, we describe how we use quantum annealing to generate test cases that can make the failure more evident and, therefore, more easy to detect.

4.2 Overview of the process

An overview of our test case generation process is shown in Figure 4. The objective is to enhance the effectiveness of test cases by identifying and modifying key areas within each input trajectory. In this context, key areas refer to specific temporal segments of the input trajectory where changes to the input values are expected to produce the largest behavioural deviations or expose potential faults in the system under test (SUT). In practice, these correspond to the time intervals where the system exhibits high sensitivity to input variations or where adequacy metrics indicate low coverage or diversity.

The process is divided into three main phases. In **Phase 1**, an initial suite of seed test cases is randomly generated and executed on the SUT, producing corresponding output trajectories. A set of adequacy metrics (such as coverage and divergence) is then extracted from both the inputs and their outputs. In **Phase 2**, these metrics are analysed to locate the most critical regions of the test cases (i.e., the key areas), which are then formulated as an optimisation problem aimed at maximising fault-revealing potential. This optimisation problem is solved using a quantum annealing algorithm, which determines the most promising regions for mutation. Finally, in **Phase 3**, the selected regions are modified through targeted mutations that are embedded back into the original test cases, producing an enhanced suite expected to achieve higher fault-detection effectiveness.

4.3 Phase 1: Generate

In Phase 1, random values are generated for data points along each test case, i.e., trajectory over time ($t \in T$), where T refers to the test duration. These values are constrained by minimum and maximum boundaries, as well as a maximum allowable rate of increase or decrease to ensure that the test case is valid. To form a smooth trajectory, these data points are connected via a piecewise non-linear curve fitting solution provided by the Matlab Optimisation Toolbox⁴. We follow this approach for all input variables to form the trajectory that represents a test case. A set of test cases forms a test suite. The test suite is then executed on the SUT to generate the output for the corresponding test case.

⁴<https://www.mathworks.com/products/optimization.html>

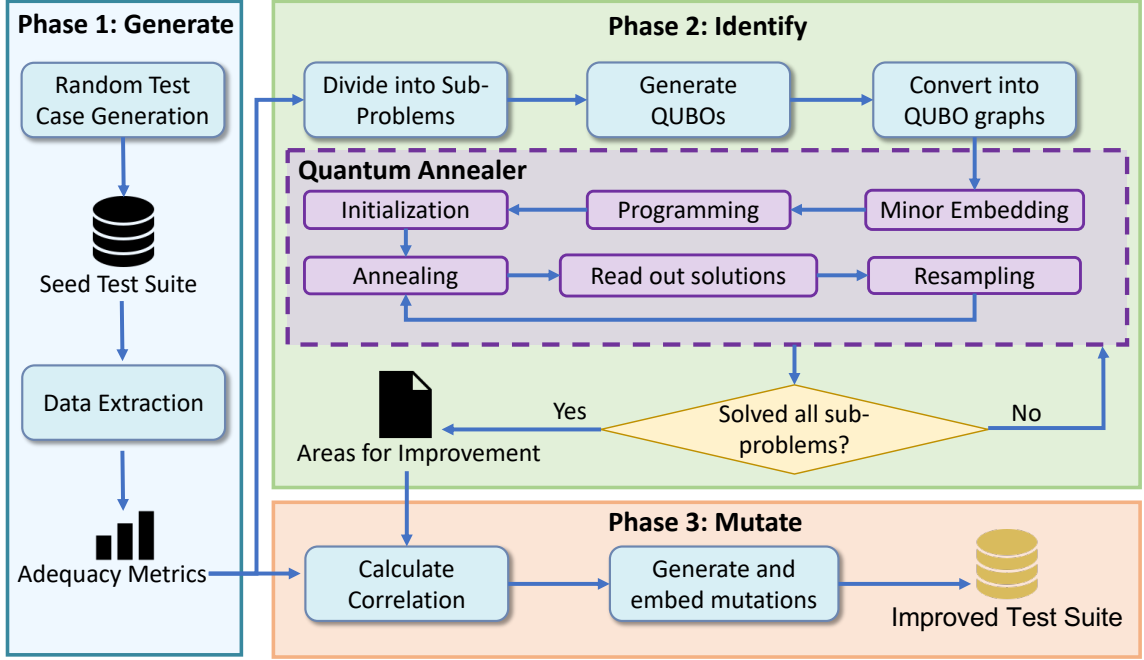


Fig. 4. Overview of the 3 phases of our test case generation process. Phase 1 generates the test cases; Phase 2 uses quantum annealing to identify regions on the test cases the can be improved; Phase 3 modifies the test cases aiming to improve it.

EXAMPLE 1. Consider our running example. Figure 5 depicts three test cases in an initial test suite and the respective outputs (expected and observed). Each test case, in practice, is essentially a series of values that, when connected, form a trajectory.

For each data point in a test case (i.e., the value of a point in time), three adequacy metrics are collected: one related to effectiveness and two related to diversity (i.e., input and output diversity) of the trajectory.

Effectiveness metric ($\mathcal{E}$): We calculate the Euclidean distance between observed output $O(t)$ (i.e., obtained from exercising the input directly on the SUT) and the expected output $E(t)$ (i.e., observed from exercising the input on a design model) at time t . In model-based testing, behavioural or design models are typically treated as reference models that describe the intended behaviour of the system under test and thus act as an abstract oracle or “ground truth” [61, 66]. The metric value for each data point $I(t)$ is then normalized using the min-max normalisation method.

$$\mathcal{E}(I(t)) = \frac{\|O(t) - E(t)\| - \mathcal{R}_{min}}{\mathcal{R}_{max} - \mathcal{R}_{min}}$$

In this formula, $\mathcal{R}_{max}$ and $\mathcal{R}_{min}$ represent the maximum and minimum values for the range of the signal.

Input diversity ($\mathcal{D}_i$) and output diversity ($\mathcal{D}_o$): Similarly, we calculate the input and output diversity for each data point in the test case. For each data point at a time t , we select a fixed range of surrounding data points (e.g., all data points within the $t \pm r$ range). This reduces the original trajectory to a slice of it (s_1). The diversity metric is then calculated by using normalised Euclidean distance between points of the input slice (s_1) and slices (within the same

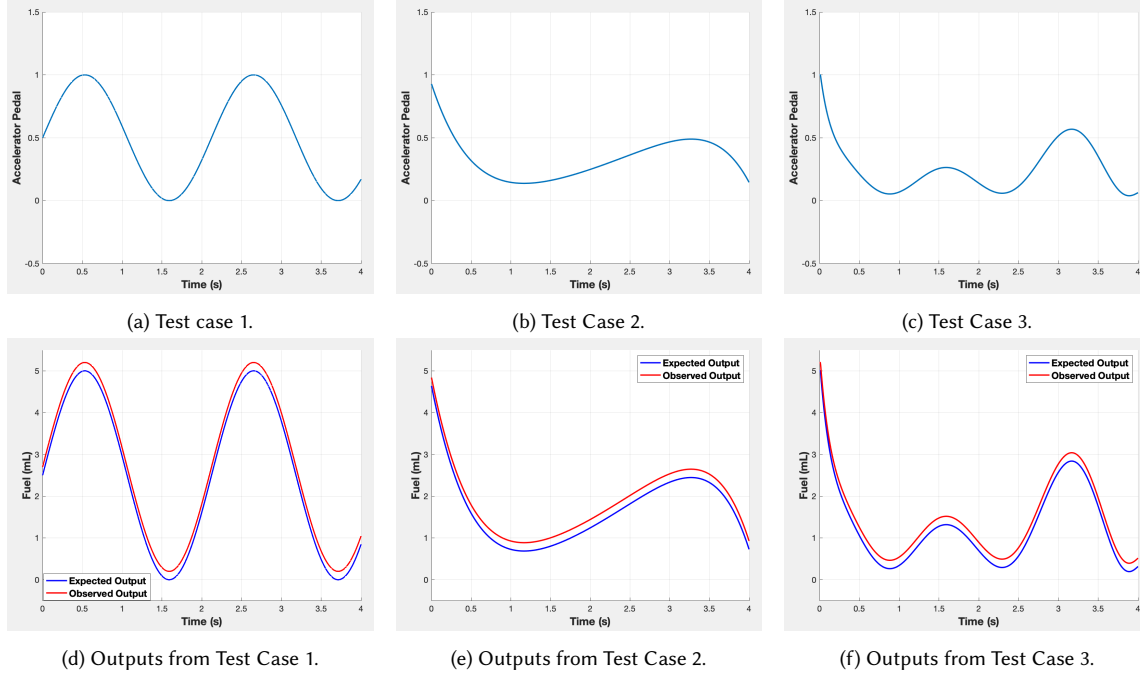


Fig. 5. Test cases in the test suite and the respective outputs from both a simulation model (expected output, in blue) and the system implementation (observed output, in red).

$[t - r, t + r]$ range) in the test suite (s_i). Similar to the effectiveness metric, the diversity formula normalises the diversity between the $[\mathcal{R}_{min}, \mathcal{R}_{max}]$ range. This is given by the formula below, which was proposed by Matinnejad et al. [49].

$$\mathcal{D}_i(t) = \frac{\sqrt{\sum_{w=(t-r)}^{t+r} (s_1(w) - s_i(w))^2}}{\sqrt{(t+r) + 1 \times (\mathcal{R}_{max} - \mathcal{R}_{min})}}$$

The input diversity for a data point is given by the minimum distance between the respective slice and the corresponding slices in the other test cases. To calculate the output diversity of a test case data point, we take a slice comprising the surrounding points within a $\pm t$ range from the output of that test case. Then, we use the same formula above to compare the slice of the respective output against slices of the same time interval for every other output. The diversity is determined by calculating the minimum distance, analogous to how we calculate input diversity.

EXAMPLE 2. Consider our running example. Table 1 shows the values for the three metrics that were computed when running the three test case. Here, we only show the values for the first 10 data points; the time difference between each data point is 0.01s.

4.4 Phase 2: Identify

In this phase, we aim to identify the data points in each test case that can be modified to improve the corresponding critical areas. To make this optimisation tractable on current quantum hardware, we first decompose the overall

Table 1. Effectiveness, input diversity and output diversity metrics collected from each test case in the running example.

<i>Index</i>	<i>Test Case 1</i>			<i>Test Case 2</i>			<i>Test Case 3</i>		
	<i>Efctvt.</i>	<i>I. Div.</i>	<i>O. Div.</i>	<i>Efctvt.</i>	<i>I. Div.</i>	<i>O. Div.</i>	<i>Efctvt.</i>	<i>I. Div.</i>	<i>O. Div.</i>
1	0.305	0.697	0.956	0.417	0.100	0.183	0.126	0.148	0.240
2	0.308	0.696	0.956	0.413	0.097	0.183	0.123	0.148	0.240
3	0.310	0.696	0.956	0.411	0.097	0.183	0.122	0.148	0.240
4	0.312	0.695	0.956	0.406	0.097	0.183	0.120	0.145	0.240
5	0.316	0.695	0.956	0.401	0.097	0.183	0.117	0.145	0.240
6	0.317	0.695	0.956	0.400	0.097	0.183	0.113	0.145	0.240
7	0.320	0.696	0.956	0.399	0.097	0.183	0.112	0.145	0.240
8	0.321	0.696	0.956	0.398	0.097	0.183	0.111	0.145	0.240
9	0.324	0.696	0.956	0.394	0.097	0.183	0.111	0.145	0.240
10	0.324	0.696	0.956	0.393	0.096	0.183	0.107	0.145	0.240

problem into smaller sub-instances, each of which can be independently solved using quantum annealing. The resulting sub-solutions are then merged to obtain the final optimisation result.

Dividing the problem into sub-problems: A QUBO problem can be represented as a fully connected graph where variables are encoded as qubits and their interactions as couplers. As the problem size increases, the number of required qubits and couplers grows non-linearly, which may exceed the capacity of current annealers and introduce noise. To address this limitation, we subdivide the optimisation into smaller, hardware-feasible QUBO sub-instances. This subdivision not only enables execution within hardware constraints but also promotes solution diversity by exploring multiple local regions of the input trajectory in parallel.

More specifically, given a test case, we first divide it into M slices (i.e., M sub-problems), and randomly sample N points from each slice. Here, the slicing is done by partitioning the total simulation time into sequential equal-sized segments (i.e., it is important to note that, because QA will be used in the next step to identify critical regions, all segments are treated with equal importance in the step.). Hence, the full trajectory i is divided into M equal-length slices, and N points are randomly sampled from each slice; sampling continues until every slice has been covered. These M subsets are used to formulate M sub-problems based on the QUBO formulation, and each sub-problem is solved separately using QA, resulting in the corresponding M sub-solutions. Eventually, the data points selected in each sub-solution are combined to form the final solution.

To determine suitable values for the number of slices (M) and the problem size (N), both of which depend on hardware constraints and the problem definition, we perform a systematic grid search (see Section 5.3). We treat M and N as hyperparameters and tune them independently, one at a time: we first fix M and sweep N over a discrete range of candidate values (from 5 to 100 data points), then fix N at its selected value and sweep M over 1 to 10 sub-problems. For each candidate configuration we run the quantum annealer and record two quantities: the QUBO fitness value (which the annealer minimises) and the resulting fault-detection rate. We select N as the value that minimises the fitness value, and M as the smallest number of sub-problems beyond which the fault-detection rate plateaus, thereby balancing effectiveness against the additional annealer run each sub-problem incurs. The bounds on this search space follow from the underlying trade-offs: N is limited by what the annealer can reliably embed (too large inflates the qubit count and adds noise, too small limits exploration), while larger M improves solution diversity across the trajectory at the cost of efficiency. Applying this procedure yields $N = 40$ and $M = 8$ (see Section 5.4), which we adopt as the default configuration throughout.

Identifying critical areas with Quantum Annealing: For each sub-problem generated in the previous steps, we construct an overall objective function in the QUBO formulation (introduced in Section 3) to select the data points. We represent the selection of each data point as a binary variable x_i in the QUBO. We set 1 to x_i if the i -th data point is selected; otherwise, we set it to 0.

This objective function integrates four objectives and a constraint into a QUBO model, using a weighted-sum strategy [76]. Specifically, three of the objectives relate to the previously introduced metrics. The fourth objective is related to avoiding selecting too many data points, thus aiming to minimise the number of selected data points. The constraint is applied to control the distance among the selected data points.

To build these three objectives, we first extract the effectiveness, input, and output diversity values of each data point in the sub-problem. Then, we minimise the distance between the sum of the metric values of the selected data points and the corresponding ideal value for that metric [76]. In detail, since we would like to maximise the sum of effectiveness values, we calculate the upper limit L^{ef} , which is the total sum of the effectiveness values for all data points. On the other hand, the minimal input and output diversity values are $L^{\text{id}} = 0$ and $L^{\text{od}} = 0$. We represent the $\text{mr} \in \{\text{ef}, \text{id}, \text{od}\}$ as the generalised metric. For each metric, the metric value of each data point is represented as v_i^{mr} . Given a trajectory with a total of N data points, the selection of all data points can be represented in a vector $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]$, where $x_i \in \{0, 1\}$ which represents whether the datapoint is or is not part of the solution.

We represent the objective function as

$$\min \text{Obj}^{\text{mr}}(\mathbf{x}) = \left(\sum_{i=0}^{N-1} v_i^{\text{mr}} x_i - L^{\text{mr}} \right)^2, \text{mr} \in \{\text{ef}, \text{id}, \text{od}\} \quad (6)$$

Since $x_i \in \{0, 1\}$, we have that $x_i^2 = x_i$. Meanwhile, we do not consider the constant terms in QUBO since they do not affect the optimisation. Thus, we can expand the objective as follows:

$$\min O^{\text{mr}}(\mathbf{x}) = \sum_{i=0}^{N-1} ((v_i^{\text{mr}})^2 - 2L^{\text{mr}} v_i^{\text{mr}}) x_i + 2 \sum_{i < j} v_i^{\text{mr}} v_j^{\text{mr}} x_i x_j, \text{mr} \in \{\text{ef}, \text{id}, \text{od}\} \quad (7)$$

Mapping to the QUBO formulation, $(v_i^{\text{mr}})^2 - 2L^{\text{mr}} v_i^{\text{mr}}$ are the linear coefficients of the variable x_i while $2v_i^{\text{mr}} v_j^{\text{mr}}$ are the quadratic coefficients between variables x_i and x_j .

EXAMPLE 3. Suppose that the sampling selected two data points ($N = 2$) from Test Case 1. The index of the selected points are 3 and 7. The vector of binary variables ($\mathbf{x}$), can have four possible values ($[0, 0]$, $[0, 1]$, $[1, 0]$, $[1, 1]$), where, for instance, $\mathbf{x} = [1, 0]$ means that the data point of index 3 is selected but the data point of index 7 is not. For $\mathbf{x} = [1, 1]$ and using effectiveness metric (ef) as the the only objective, replacing the values in Equation 7 leads to the following:

$$\begin{aligned} O^{\text{ef}}([1, 1]) &= \sum_{i=0}^{2-1} ((v_i^{\text{ef}})^2 - 2L^{\text{ef}} v_i^{\text{ef}}) x_i + 2 \sum_{i < j} v_i^{\text{ef}} v_j^{\text{ef}} x_i x_j \\ &= (.31^2 - 2 * .63 * .31 * 1) + (.32^2 - 2 * .63 * .32 * 1) + 2 * (.31 * .32 * 1 * 1) \\ &= -0.398 \end{aligned}$$

On the other hand, $O^{\text{ef}}([0, 0]) = 0$, $O^{\text{ef}}([1, 0]) = -0.294$, and $O^{\text{ef}}([0, 1]) = -0.3$, which indicates that, when it comes to effectiveness only, selecting both data points results in the optimal value.

In the fourth objective, similarly, we minimise the distance between the total number of selected data points in the sub-problem and the theoretical lower limit $L^{\text{num}} = 0$. We create one more objective:

$$\min \text{Obj}^{\text{num}}(\mathbf{x}) = \left(\sum_{i=0}^{N-1} x_i - L^{\text{num}} \right)^2 \quad (8)$$

We can get the linear and quadratic coefficients of corresponding variables in the QUBO formulation similar to the above. With the four objectives, we can combine them into an overall QUBO formulation with the weighted-sum method. Hence, we assign a weight to each metric ($w_{\text{mr}} \mid \text{mr} \in \{\text{ef}, \text{id}, \text{od}\}$) as well as to the objective on the optimal number of points returned w_{num} , which gives us:

$$\min O(\mathbf{x}) = \sum_{\text{mr} \in \{\text{ef}, \text{id}, \text{od}\}} w_{\text{mr}} O^{\text{mr}}(\mathbf{x}) + w_{\text{num}} O^{\text{num}} \quad (9)$$

EXAMPLE 4. Consider that the sampling selected three data points ($N = 3$) from Test Case 2. The index of the selected points are 1, 4, and 7. For this example, we consider all possible objectives (ef, id, od, num) and the weights are set to $w_{\text{ef}} = 0.25$, $w_{\text{od}} = 0.125$, $w_{\text{id}} = 0.125$, $w_{\text{num}} = 0.5$. The vector of binary variables ($\mathbf{x}$), can have eight possible values ($[0, 0, 0]$, ..., $[1, 1, 1]$). Therefore, according to Equation 9, we have that:

$$\begin{aligned} O([0, 0, 0]) &= 0, & O([0, 0, 1]) &= -0.162 \\ O([0, 1, 0]) &= -0.131, & O([0, 1, 1]) &= -0.279 \\ O([1, 0, 0]) &= -0.113, & O([1, 0, 1]) &= -0.208 \\ O([1, 1, 0]) &= -0.259, & O([1, 1, 1]) &= -0.265 \end{aligned}$$

Hence, the choice of two data points (index 4,7) yields the optimal result.

We note that in Example 4 the weights are set to $w_{\text{ef}} = 0.25$, $w_{\text{od}} = 0.125$, $w_{\text{id}} = 0.125$, and $w_{\text{num}} = 0.5$. These values were empirically determined to balance the relative influence of the optimisation terms: greater weight is assigned to w_{num} to prioritise the selection of an adequate number of data points (ensuring sufficient exploration of the test case without completely modifying it), while the remaining weights were tuned to give comparable contributions from fault-effectiveness (w_{ef}) and divergence-related terms (w_{od} , w_{id}). This configuration was found to produce stable convergence across two case studies without overemphasising any metric.

In the final step of the process (Phase 3, where we mutate the trajectory), we modify the area surrounding each data point selected by the quantum annealer. If two selected data points in the solution are too close to each other, their corresponding mutation regions may overlap, which can reduce the overall effectiveness of the mutation process. To prevent this, we introduce a constraint enforcing a minimum temporal separation between selected points. Specifically, for each pair of variables x_i and x_{i+1} , they cannot both be set to 1 if their timestamps are within a minimum temporal distance of 2 seconds.

Since the objective function is formulated as a minimisation problem, we incorporate a penalty term to discourage such violations. The penalty value P is used to make simultaneous selection of temporally adjacent points significantly more costly, thereby enforcing spacing between mutations:

$$\min C(\mathbf{x}) = P \cdot x_i x_{i+1} \quad (10)$$

Here, P acts as a hard constraint weight. Its magnitude must be sufficiently large relative to the other coefficients in the cost function to ensure that proximity violations are strongly penalised and never favoured by the optimiser. In practice, P is set an order of magnitude higher than the largest optimisation term, effectively guiding the annealer to prioritise spatial separation between selected data points.

Finally, the overall objective function is shown below. We can merge all the linear terms and coefficient terms to create the overall QUBO formulation.

$$\min Obj(\mathbf{x}) = \sum_{mr \in \{cr, id, od\}} w_{mr} Obj^{mr}(\mathbf{x}) + w_{num} Obj^{num} + C(\mathbf{x}) \quad (11)$$

Once all of the sub-solutions have been found, they are merged to compose the final problem and given to the quantum annealer. In summary, the quantum annealer finds the optimal data points for each sub-problem and then finds the optimal data points out of those to determine the final solution.

While there could be other possible implementations of QUBO formulations, we adopt the above formulation because our goal is to construct a practical and accessible testing approach based on QA that enables a fair comparison with classical optimization techniques. Incorporating other domain knowledge, such as assigning non-uniform importance to different regions, would introduce additional layers of complexity and would require analogous extensions to classical baselines to ensure fairness. As the purpose of this work is not to develop the most sophisticated testing framework, but to evaluate the feasibility and performance of QA in this context, we retain the simpler, uniform formulation.

4.5 Phase 3: Mutate

In this phase, we first generate the mutations, and then we embed them into the existing test cases.

Generating mutations: The purpose of mutation is to enhance the effectiveness of test cases around the data points identified in Phase 2. Using the effectiveness metrics collected in Phase 1, mutations are applied to the selected data points. For each identified data point $I(t)$, we define a surrounding temporal range and compute a *correlation value* $c(t)$ that quantifies the linear relationship between the input value $I(t)$ and the corresponding effectiveness value $\mathcal{E}(t)$. In this context, the correlation measures how strongly variations in the input influence the observed effectiveness over time. The correlation values are normalised to the interval $[-1, 1]$ and used to guide the direction and intensity of the mutation:

$$I(t)_{mut} = I(t)^{(1-c(t))} \times (\mathcal{R}_{max} - \mathcal{R}_{min}) + \mathcal{R}_{min}$$

where $\mathcal{R}_{max}$ and $\mathcal{R}_{min}$ denote the upper and lower limits of the input range. A positive $c(t)$ biases the mutation towards $\mathcal{R}_{max}$, increasing the input value, while a negative correlation biases it towards $\mathcal{R}_{min}$, decreasing it. The magnitude of $c(t)$ controls the mutation intensity, with values closer to ± 1 leading to stronger perturbations.

Embedding mutations: Once mutations are generated, they are incorporated into the original test cases by modifying the temporal region surrounding each mutated data point. This ensures a smooth transition between the original and mutated trajectories, as illustrated in Figure 6. The embedding is achieved using a curve-fitting regression technique (similar to the one applied in Phase 1) to interpolate between neighbouring points, preserving the overall continuity of the input signal. Importantly, all mutated test cases adhere to the same validity constraints established during generation,

including input range limits and the maximum rate of change. These constraints guarantee that the modified trajectories remain physically and semantically consistent with the system assumptions introduced in Section 4.3.

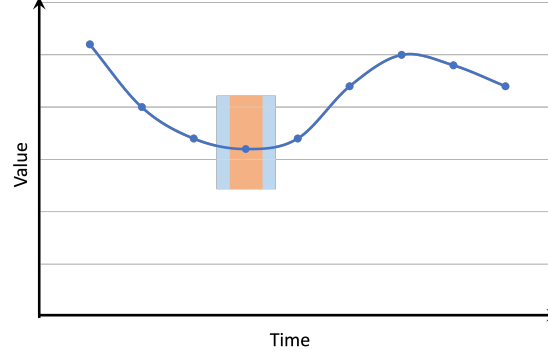


Fig. 6. Target area of a test case mutation.

The output of Phase 3 is an enriched test suite with additional test cases that are modified to target and enhance fault detection effectiveness through mutation-driven adaptations.

EXAMPLE 5. Consider our running example. Figure 7 depicts the mutated input without (Figure 7a) and with curve fitting (Figure 7c). The respective outputs are also shown in this figure (Figures 7b and 7d, respectively). Note how the distance between the expected and observed behaviour is more prominent compared to when the non-mutated input was used (Figure 5f), which makes faults easier to detect.

5 CONTROLLED EXPERIMENT

In this section, we present the application of the strategy proposed in Section 4 to two case studies: an autonomous vehicle platoon and a pneumatic suspension system. We first outline our research objectives (Section 5.1), then we provide a description of the case studies (Section 5.2), followed by the methodology definition (Section 5.3.1) and lastly, the results and discussion (Section 5.4).

In our methodology, we use system specifications developed in Simulink [30] - a widely-used development and simulation language for CPS - as well as variants in which faults have been manually inserted. We generate test suites using our process (described in Section 4) and use their outputs to reach a verdict on how many of the inserted faults have been detected. The verdict is given by a conformance notion that, for the same input, checks that the distance between the outputs from the correct and faulty models is not greater than a maximum threshold ϵ .

5.1 Research objectives

The main goal of our evaluation is to explore (in terms of efficiency and effectiveness) the use of quantum annealing for test case generation. To this end, we consider two case studies that manifest safety-critical hazards (such as vehicular collisions) that require rigorous verification. We first explore the use of quantum annealing across different problem sizes and number of sub-problems to find optimal configuration values; we then use those values to reach our conclusions on the effectiveness of the generated test suite and the efficiency to generate test suites. Based on this, we aim to answer the following research questions.

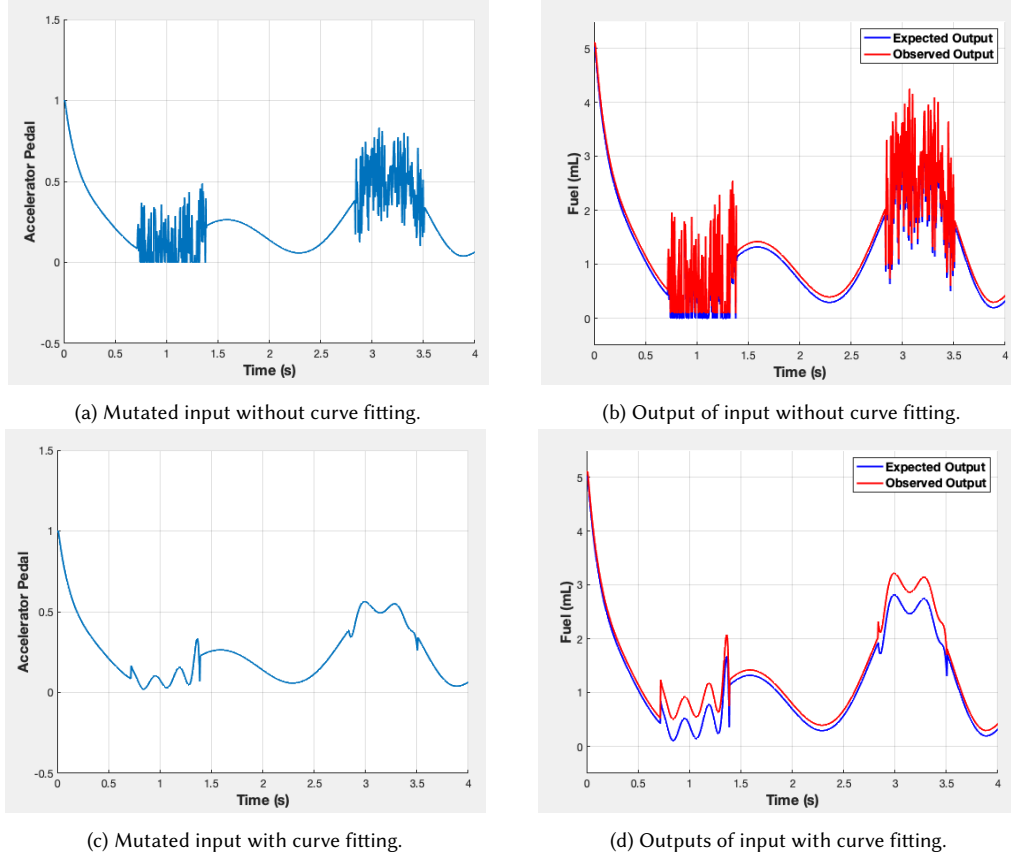


Fig. 7. Mutated Test Case 3 (taken from the running example) and its outputs (expected and observed). The curve fitting solution turns raw data points (a) into a smoother curve that represents more realistic CPS behaviour (c).

5.1.1 Baselines. We compare our approach against test suites generated from two state-of-the-art alternative heuristics, as well as a random test case generation approach. As one of the baseline approaches, we use the classical counterpart to quantum annealing (namely, simulated annealing [13]) and, additionally, as the second baseline, we use NSGA-II [28]. We have used the D-Wave API ⁵ to implement the simulated annealing approach and the Pymoo API [14] to implement the NSGA-II approach. Both heuristics are used to solve the same optimisation problem from Phase 2. In both the simulated annealing and the NSGA-II cases, we employ a single function that combines multiple objectives as one objective ($\min f(x) = w_1 f_1(x) + \dots + w_n f_n(x)$) where we have explored different weights to approximate a Pareto front.

Lastly, as a third alternative approach, we randomly select areas of the inputs to be modified. The first two approaches are suitable alternatives to quantum annealing, and the latter is akin to standard mutation-based fuzzing [52].

5.2 Case Studies

In this section, we describe the two case studies used throughout our empirical evaluation. The selected case studies were chosen because they exhibit key characteristics of cyber-physical systems (CPS) that make testing particularly

⁵https://docs.ocean.dwavesys.com/en/stable/docs_neal/reference/sampler.html

challenging: hybrid dynamics combining discrete and continuous behaviours, complex interactions across subsystems including internal and external communication, and human-in-the-loop interaction where operator inputs shape system behaviour. In addition, the models feature a large input space which warrants the need for search procedures such as simulated annealing in their testing. These features ensure that the chosen cases are representative of realistic CPS while remaining suitable for evaluation.

5.2.1 Platoon of connected vehicles. Vehicular platooning is a cooperative and autonomous driving technology that connects two or more vehicles into a convoy. The primary objective of the convoy is to maintain a close yet safe distance between vehicles using vehicle-to-vehicle (V2V) communication and automated driving technologies [12]. This study utilises an open-source model of the system, originally developed for a software verification study [5]. The system aims to ensure that all follower vehicles in the platoon stay behind the lead vehicle, maintain safe distances from each other, and remain within communication range, as illustrated in Figure 8.

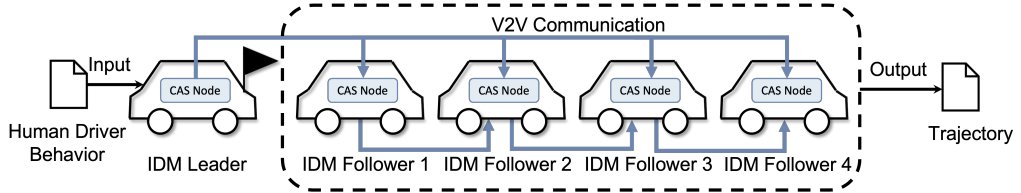


Fig. 8. Diagram of a platoon of connected vehicles [46]. The input is the human behaviour and the output is the followers trajectory. The leading vehicle transmits its data to all followers, and each follower transmits their data to the vehicle directly behind it.

To enable follower vehicles to accelerate and decelerate automatically, an Intelligent Driver Model (IDM) controller [73] is employed. The IDM is a continuous car-following model that describes how a vehicle adjusts its acceleration based on its own velocity, the velocity difference to the preceding vehicle, and the current inter-vehicle distance. It captures the trade-off between maintaining a desired speed and keeping a safe following distance. Each vehicle is also equipped with a communication node responsible for transmitting data packets. The model focuses solely on longitudinal vehicle movement, assuming the platoon operates along a straight, extended highway without significant directional changes. The main input for the testing campaign is the behaviour of the human driver in the lead vehicle, specifically their acceleration pattern. Once this input is provided, V2V communication ensures that the follower vehicles autonomously adjust their behaviour to align with the lead vehicle. The system's output is a trajectory that captures the acceleration of the follower vehicles.

5.2.2 Suspension System. In this case study, we analyse an automotive pneumatic suspension system [57], designed to enhance driving comfort by maintaining the chassis level and compensating for road irregularities. The system achieves this by utilising a suspension mechanism that connects valves at each wheel to a central compressor and an escape valve (see Figure 9a).

The primary objective is to maintain the chassis level as close as possible to a pre-defined set point for each wheel. Adjustments are guided by tolerance intervals specific to each wheel, as illustrated in Figure 9b. These intervals are defined as $[sp - otl, sp + otu]$ for the outer range and $[sp - itl, sp + itu]$ for the inner range, where sp denotes the set point, or the target chassis level. The variables itl , itu , otl , and otu represent the lower and upper thresholds for the inner and outer tolerance intervals.

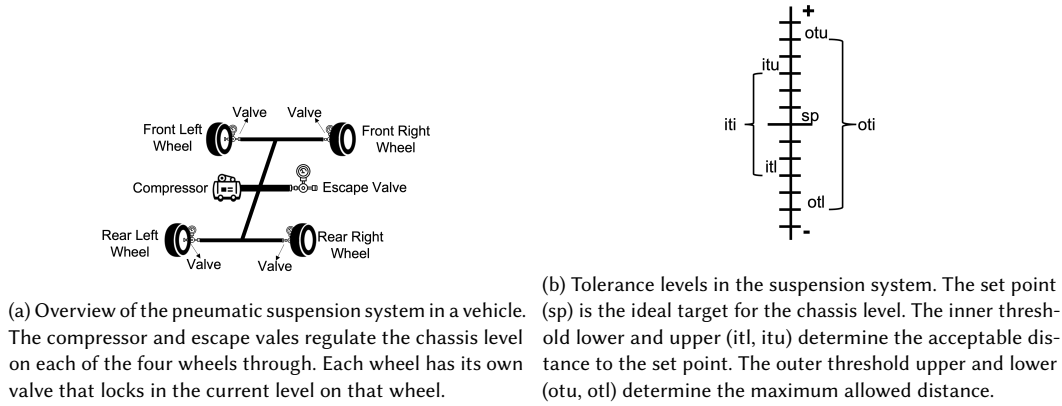


Fig. 9. Suspension system.

The system processes four inputs: (i) road disturbance levels, (ii) adjustments to the chassis level made by the compressor, (iii) adjustments made by the escape valve, and (iv) an indicator signalling whether the vehicle is turning, during which, pneumatic adjustments are restricted. The system's output is the current chassis level, which reflects its continuous effort to stabilise and optimise ride comfort.

5.2.3 Modelling. The platooning and the suspension systems are implemented in Matlab/Simulink. The platooning model comprises 36 blocks arranged across three hierarchical levels, including subsystems for vehicle dynamics, inter-vehicle communication, and the Intelligent Driver Model controller. Its input (the lead-vehicle acceleration) vary within the ranges $[-3, 3]$ and we apply standard IDM ranges for the followers vehicles [73]. The simulation is executed for 30 seconds using a fixed-step solver with a step size of 1 ms, and input trajectories between mutation points are interpolated using linear interpolation.

As for the pneumatic suspension system, its model comprises 21 blocks organised over two levels, modelling the valve dynamics, and the chassis-level control. Its inputs include road-disturbance profiles in the range $[-20, 20]$ cm and the valve-actuation commands as well as the flag for vehicle turns are in $[0, 1]$. This model is also simulated for 30 seconds with a fixed-step solver at 1 ms resolution, again applying linear interpolation between mutated points.

5.3 Experiment Design

In this section, we explain the experiment design. Particularly, we describe the methodology (Section 5.3.1), and our metrics and hypotheses (Section 5.3.2).

5.3.1 Methodology. An overview of the experiment methodology is as follows. First, we insert faults into our extensively tested and (believed-to-be) correct system model. This is done by inserting one fault in the model at a time and, thus, generating a faulty variant. Then, we employ quantum annealing as well as the three alternatives to generate the test suites. More specifically, the decision of which regions of the test cases to be mutated (i.e., Phase 2) is done by four heuristics: (i) quantum annealing, (ii) simulated annealing, (iii) NSGA-II, and (iv) a random-based decision. To obtain statistical significance, the test suite generation and execution parts of the experiment are repeated 10 times, followed by a statistical analysis of the results which is used to answer the RQs.

For context, the classical algorithms were executed on a 2019 MacBook Pro that has a 2.6 GHz 6-Core Intel Core i7 CPU, AMD Radeon Pro 5300M 4 GB GPU, and 16 GB 2667 MHz DDR4 memory. As for the quantum annealer, all experiments were executed on a D-Wave Advantage system, which implements quantum annealing on a Pegasus topology comprising more than 5000 qubits. For each QUBO instance, we configured the solver using parameters recommended in the D-Wave documentation: an annealing time of 20 μ s, 500 reads per problem instance, a chain strength set to 1.5 times the maximum absolute bias to minimise chain breaks, and the default flux-bias settings.

Generating faulty variants: We employ the Simulink mutation tool FIM [10] in our method to insert faults. We insert five types of faults: the *Delay Operator* simulates the introduction of delays, while the *Noise Operator* adds noise to signals within a Simulink model. The *Package Drop* operator replaces a variable’s value with a different one. Additionally, the *Logical Operator Replacement* and *Arithmetic Operator Replacement* mutators substitute an operator with another of the same type. These mutation operators were selected based on their applicability to our models. In total, we introduced 10 faults of each type into each case study. We avoided inserting multiple faults simultaneously, as this could lead to unpredictable interactions, such as one modification cancelling out the effect of another.

Generating test suites: Next, we generate the test suites: an initial test suite (i.e., obtained from Phase 1) and four mutated versions produced using quantum annealing and the baseline alternatives.

To answer RQ1, we compare the results of varying configurations of quantum annealing across different problem sizes and numbers of sub-problems. Here, a configuration refers to the set of annealing parameters and solver options that influence the search dynamics of the quantum annealer, including the annealing time, the embedding strength between logical and physical qubits, and the scheduling parameters that define the anneal path.

We first investigate the effect of varying the problem sizes (i.e., the number of data-points given to the annealer). Then, we investigate how different numbers of sub-problems can affect the results, by splitting the test case into a varying number of sub-regions, from one all the way up to ten sub-problems. Lastly, we have a look at the scheduling parameter: in our pilot experiments, the number of samples (problem size) and the anneal scheduling have the most effect on the final result. We report the results on these three aspects below.

To answer RQ2 and RQ3, we compare the results between the test suite generated from the quantum annealing approach versus the ones generated by the alternative approaches. When using quantum annealing, we have made use of the results from RQ1 to configure the optimal problem size and number of sub-problems. When using the alternative approaches, unlike with quantum annealing, there’s no restriction to the number of variables and, hence, we do not divide the problem into sub-problems or make use of sampling. Instead, the entire trajectory is used by the baselines to decide on the optimal candidates.

To remove bias, we use the same weights and objectives (namely, maximising criticality, minimising input and output diversity and the number of points selected) as well as the proximity constraint. Although it may sound unintuitive, in our search for the most appropriate points for mutation, we search for the points in the *existing* test-suite with the **minimum** diversity, because by mutating them, we are likely to improve the diversity in the *output* test-suite. The weights are set to 0.25 for criticality, 0.125 for input diversity, 0.125 for output diversity, and 0.5 for number of data-points. In the case of quantum annealing, the weight for the proximity constraint is set to 10. The choice of weights is based on a set of pilot experiments to determine these values. We experimented on a range of values and across multiple case studies to empirically maximise fault detection rate. In particular, we note that in order to set a soft constraint (i.e., for the proximity objective), the value for the weight must be orders of magnitude higher than that of the other objectives. It has been shown that values that are too small will lead to infeasible solutions returned by the solver while values that are too large may lead to slower convergence; the recommended value is at least 10x

that of the highest weight [9]. With respect to the other metrics, greater weight is assigned to w_{num} to prioritise the selection of an adequate number of data points (ensuring sufficient exploration of the test case without completely modifying it), while the remaining weights were tuned to give comparable contributions from fault-effectiveness (w_{ef}) and divergence-related terms (w_{od} , w_{id}). This configuration was found to produce stable convergence across case studies without overemphasising any single metric.

Gathering the results: We execute the test suites on our systems to collect the outputs, which are given to a test oracle to determine the verdicts. More specifically, given one of the test suites, we execute it on the correct model and on a faulty variant and check whether the fault is detected by comparing the outputs. We execute this step for each combination of (test suite $\times$ faulty variant) to calculate how many faults each test suite detects. As is the case with test cases, the outputs of this execution are also represented by trajectories. Thus, input and output trajectories capture the history of input and output variables throughout the execution of the system, respectively. We say that a test suite has identified an inserted fault if, when executed on the system where that fault was inserted, one of its test cases yields a fail verdict.

5.3.2 Metrics and Hypotheses. In this experiment, we hypothesise about the correlation between problem size and number of qubits and, additionally, about the efficacy and efficiency of quantum annealing. In terms of metrics, we collect the average percentage of faults detected (FD) by each approach as well as the time taken to generate the test suite. These metrics essentially quantify effectiveness in terms of the fault detection rate and effectiveness in terms of test case generation time. Essentially, the fault detection rate is defined as the number of faults detected by the test suite as a whole, out of the total inserted. A fault is considered detected if at least one test case in the suite yields a fail verdict when the suite is executed on the corresponding faulty variant. We also collect the average final fitness function value (FV), which is calculated based on the minimisation of the objective function of the problem (i.e., QUBO formulation). We have defined one hypothesis for each of our research questions and they are presented in Table 2.

Table 2. Hypotheses devised for our experiment. Hypothesis A evaluates the scalability of quantum annealing systems, and hypothesis B and C evaluate their effectiveness and efficiency compared to the state of the art.

Hypothesis A		Hypothesis B		Hypothesis C	
H_{A0}	$f(p)$ is linear	H_{B0}	$PFD_{qt} \leq PFD_{alt}$	H_{C0}	$Time_{qt} \geq Time_{alt}$
H_{A1}	$f(p)$ is non-linear	H_{B1}	$PFD_{qt} > PFD_{alt}$	H_{C1}	$Time_{qt} < Time_{alt}$

For RQ1, where we hypothesise about the correlation between the number of qubits and the problem size, we have devised Hypothesis A; the null hypothesis (H_{A0}) states that the number of physical qubits (given by $f(p)$) employed by the quantum annealer grows linearly with the problem size (p). This experiment aims to refute such a hypothesis. Thus, an alternative hypothesis (H_{A1}) is also defined, which can be accepted in case its counterpart is rejected.

With respect to RQ2, which focuses on improvements made by the process with quantum annealing, we compare the fault detection rate of the quantum annealing test suite (PFD_{qt}) against that of the alternative test suites (PFD_{alt}). We hypothesise that that quantum annealer is more effective than the alternatives and, hence, we also aim to refute the null hypothesis. Analogously, to answer RQ3, we have defined Hypotheses C, which compares the time taken to generate the test cases.

5.4 Results

In this section, we present and discuss the results of our experiment. We answer each research question by providing concrete data and offering insights about the use of quantum annealing. A brief conclusion of our findings can be found in Section 5.4.5.

It is important to note that quantum annealing may occasionally return suboptimal solutions, which can reduce fault-detection effectiveness compared to classical baselines. This behaviour accounts for part of the performance gap observed between approaches. In our evaluation, we use effectiveness (fault-detection rate) as a practical proxy for optimality, since achieving the global minimum in the optimisation process is reflected in higher detection capability. With the chosen configuration values (e.g., scheduling, problem size, and embedding settings) we observe that the distance between the solutions returned by the annealer and the global optima found by classical methods remained minimal (around 5% - see below). Overall, this phenomenon is not unique to quantum annealing but rather an inherent characteristic of heuristic optimisation algorithms, which naturally trade global optimality guarantees for computational tractability. Consequently, the impact of suboptimality in our case studies is limited and well within expected bounds for heuristic-based approaches.

5.4.1 RQ1: When using quantum annealing, what is the correlation between problem size, the number of physical qubits, and effectiveness of the results? To address this question, we executed the quantum annealer with varying numbers of data points, collected the metrics, and generated mutated test suites as outlined in our process. These suites were then executed against faulty variants of our case studies to compute fault detection rates. Table 3 summarises the results and displays the average values of the number of physical qubits (NQ) employed after the embedding step, the final fitness function value (FV), the fault detection rate (FD) of the test suite, and the time taken in seconds. Importantly, to collect these metrics, the fault detection rates were computed using data points from the entire trajectory without dividing the problems into sub-problems.

Problem Size	Suspension				Platooning			
	NQ	FV	FD	Time	NQ	FV	FD	Time
5	11.7	0.2885	54.2	0.19	13.5	0.3185	48.3	0.19
10	18.1	0.2997	54.4	0.29	19.4	0.3342	49.5	0.37
15	33.3	0.3212	56.6	0.39	29.5	0.3271	46.1	0.69
20	54.8	0.2148	58.7	0.64	53.9	0.3109	47.3	0.84
25	92.3	0.2113	56.3	0.68	91.1	0.3055	41.4	1.19
30	125.9	0.2105	61.0	0.99	127.0	0.2980	49.8	1.26
35	160.4	0.2013	64.2	1.01	160.0	0.3011	52.3	1.49
40	205.3	0.2198	68.9	1.19	198.2	0.2901	52.1	1.86
45	287.8	0.2476	69.7	1.36	281.4	0.3683	52.0	2.08
50	326.7	0.3124	57.8	1.59	328.9	0.3792	51.9	2.22
60	465.1	0.3075	56.3	1.78	466.2	0.3772	51.9	2.79
70	649.6	0.3689	58.0	2.10	637.9	0.3801	50.8	3.15
80	912.4	0.3591	53.2	2.47	922.5	0.3792	51.5	3.26
90	1025.6	0.3871	56.9	2.67	1046.8	0.3859	52.5	4.09
100	1282.7	0.3982	65.2	3.01	1271.2	0.3878	53.9	4.57

Table 3. Results collected to answer RQ1. For each case study (suspension and platooning), we increase the problem size and show the respective number of qubits (NQ), final value (FV), fault detection rate (FD), and time taken when running the quantum annealer.

We use the minimisation of the QUBO function value (FV) as the primary criterion for defining the optimal problem size since it directly corresponds to the objective optimised by the quantum annealer and provides a consistent measure. The fault detection rate (FD) on the other hand, while it directly correlates with the function value (as per Table 3), can show variations due to stochastic sampling and model-inherent randomness. As for the Time metric, it trivially decreases with smaller problem sizes but at the expense of reduced exploration and, therefore, lower effectiveness, making it unsuitable as an optimisation target.

For these reasons, FV remains the most reliable and theoretically grounded criterion for selecting the optimal configuration. Table 3 indicates optimal problem sizes of 35 and 40 for the suspension and platooning systems, respectively. Consequently, we selected 40 as the default problem size ($N = 40$) for subsequent experiments.

In order to answer Hypothesis A, as shown in Table 4, we have observed that the number of physical qubits (NQ) increases non-linearly with the problem size for both case studies. Although the logical problem size grows linearly by construction, the mapping of QUBO variables onto the physical hardware requires embedding into a connectivity graph of the annealer. This process introduces qubit chains that represent logical connections between variables, leading to a non-linear growth as the problem becomes more connected and complex.

Table 4. Number of qubits (NQ) and the growth (Δ NQ) for each problem size across both case studies (suspension and platooning).

Problem Size	Suspension NQ	Δ NQ (Suspension)	Platooning NQ	Δ NQ (Platooning)
5	11.7	–	13.5	–
10	18.1	+6.4	19.4	+5.9
15	33.3	+15.2	29.5	+10.1
20	54.8	+21.5	53.9	+24.4
25	92.3	+37.5	91.1	+37.2
30	125.9	+33.6	127.0	+35.9
35	160.4	+34.5	160.0	+33.0
40	205.3	+44.9	198.2	+38.2
45	287.8	+82.5	281.4	+83.2
50	326.7	+38.9	328.9	+47.5
60	465.1	+138.4	466.2	+137.3
70	649.6	+184.5	637.9	+171.7
80	912.4	+262.8	922.5	+284.6
90	1025.6	+113.2	1046.8	+124.3
100	1282.7	+257.1	1271.2	+224.4

In both the Suspension and Platooning systems, the number of qubits escalates rapidly beyond a problem size of 25, confirming that the relation between problem size and NQ is non-linear.

To further refine the approach, we analyse the effect of dividing the problem into different numbers of sub-problems. Using a fixed problem size of 40, we split the input trajectories into varying number of slices and run the quantum annealer on each slice. Each run on each slice returns a set of 3 to 4 points. We then combine all of this point into a single set and run the annealer one more time to obtain the final solution. Essentially, we merge the sub-solutions to form the final solution.

Results, depicted in Figure 10, revealed consistent patterns across both case studies: fault detection rates improved with increasing sub-problems, plateauing around six sub-problems. Beyond this point, the results exhibited reduced variance, indicating greater consistency. However, increasing the number of sub-problems incurs additional computational overhead due to repeated quantum annealer runs. Based on these observations, we select 8 sub-problems ($M = 8$) as the

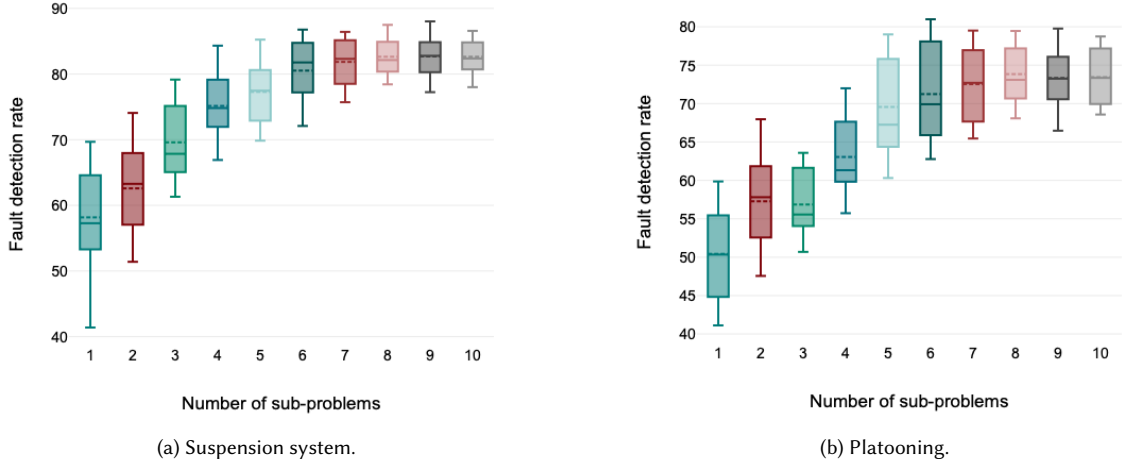


Fig. 10. Finding the optimal number of sub-problems by analysing fault detection rates (the higher the better). Increasing the number of sub-problems has diminishing returns and plateaus beyond 7 sub-problems. Results are shown across both case studies (suspension and platooning).

optimal number that balances effectiveness and efficiency for all future executions of this experiment. As shown in Figure 10, 8 sub-problems is when the curve hits a plateau and we observe negligible effectiveness gains but lower efficiency.

The goal of RQ1 is to investigate how varying problem sizes affect the number of physical qubits required by the annealer. The precise values obtained for the number of qubits employed are influenced by the configuration and characteristics of the case study under consideration, which limits the generalisability of our findings. The observed trend, however, is consistent with prior work in the literature [76], which also reports a very similar growth trend in qubit number with problem size. We acknowledge the threat to validity, and note that further case studies are needed to strengthen external validity and confirm the broader applicability of these results.

5.4.2 Scheduling. We ran tests on the D’Wave quantum annealer using our case studies to find out how different configurations for pauses (i.e., the annealing scheduling) affect the results. The outcome is shown on Figure 11. The results of our experiments show that introducing short, microsecond-scale pauses (up to approximately 500 μs) near the *middle-to-late* stages of the annealing process leads to measurable improvements in solution quality. Both the *timing* and the *duration* of a pause play a critical role in influencing the system’s ability to relax toward lower-energy states.

Pauses introduced too early in the anneal (e.g., near 10% or 20% of the annealing time) occur when the initial Hamiltonian H_{initial} still dominates, meaning the system remains largely in superposition state. Pausing in this situation seems to have little effect because there are no well-defined energy minima to settle into. Similarly, pausing too late in the anneal (e.g., $s > 0.9$) is also ineffective, as the problem Hamiltonian H_{problem} already dominates and the system is almost out of superposition and locked into the final state and relaxations are no longer warranted.

Regarding pause duration, the results suggest that longer pauses (beyond several hundred microseconds) provide diminishing returns. This can be explained by the fact that these quantum systems can reach equilibrium relatively quickly. Once this equilibrium is achieved, extending the pause does not further improve the probability of finding the ground state, but instead increases total runtime without additional benefit. Therefore, the most effective strategy is to

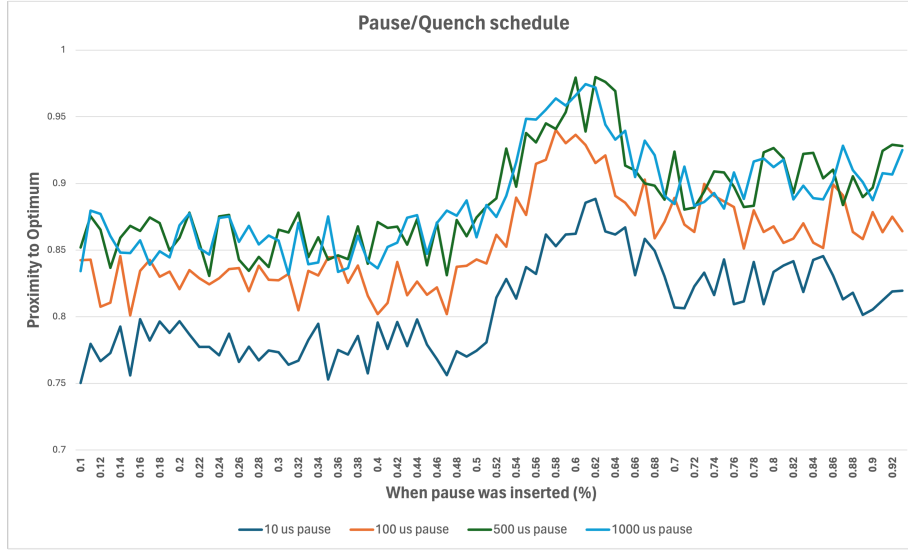


Fig. 11. Improving results by inserting micro pauses in the annealing schedule. The length of the pause (shown across different colours) and when to insert the pause (shown across the x axis) affects the proximity to the optimum (shown across the y axis).

apply short, well-timed pauses towards the middle-end point of the annealing process; we applied our learnings to the remaining of this experiment.

5.4.3 RQ2: Is using quantum annealing to mutate the test cases more effective compared to the alternatives? To evaluate the effectiveness of quantum annealing (QA), we compared its fault detection rates (of default QA and with modified schedule) against those achieved by simulated annealing (SA), NSGA-II, and random selection using the optimal configurations established in RQ1. Figure 12 presents the results for both case studies.

The median fault detection rates for all three heuristics were similar. For the suspension system, quantum annealing achieved a median fault detection rate of 82.6% (± 2.81), compared to 82.8% (± 2.23) for simulated annealing and 83.0% (± 1.71) for NSGA-II. For the platooning system, the respective rates were 73.0% (± 3.02), 73.7% (± 2.56), and 74.2% (± 2.47). To assess whether these observed differences were statistically significant, we conducted pairwise two-tailed Student's *t*-tests [72] with a confidence level of 5% ($\alpha = 0.05$). In all cases, the resulting *p*-values were greater than 0.05, indicating that the mean differences among the heuristics were not statistically significant. Even then, simulated annealing and NSGA-II exhibited lower variance. The slightly higher dispersion observed for quantum annealing is likely due to hardware-level noise, stochastic effects in the annealing process, and variability introduced by the embedding of logical qubits onto physical hardware. These factors can cause fluctuations in solution quality across runs, particularly when problem embeddings are large. Nevertheless, the central tendency of the results remains comparable, indicating that such variability does not materially affect overall fault-detection effectiveness. Lastly, Random selection performed significantly worse than all heuristic approaches, underscoring the benefit of structured optimisation techniques.

5.4.4 RQ3: Is using quantum annealing to mutate the test cases more efficient compared to the alternatives? Considering that our strategy for using quantum annealing involves dividing the problem into sub-problems and merging them at the end, we have taken this time into account on the data below. Hence, efficiency comparisons considered the time

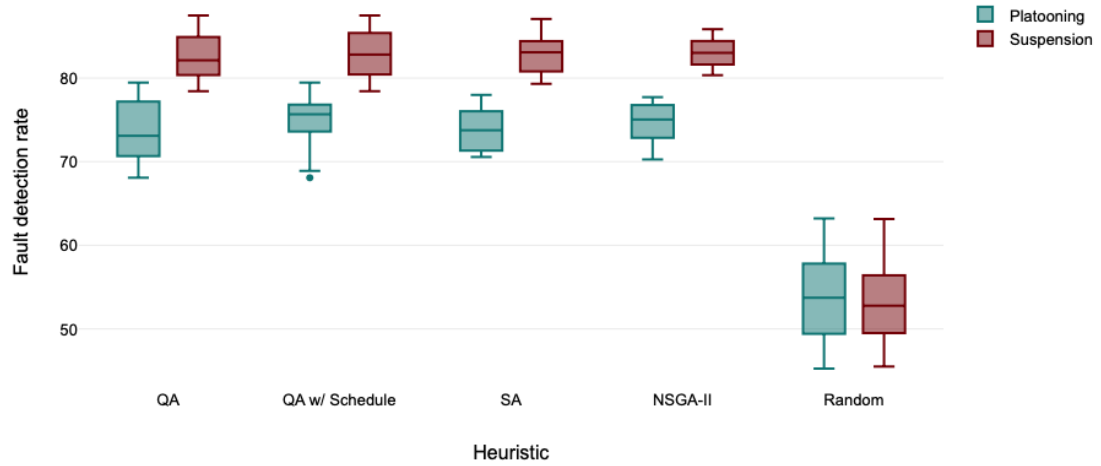


Fig. 12. Fault detection rate by heuristic. Quantum Annealing (QA) and Quantum annealing with modified schedule (QA w/ Schedule) show similar effectiveness compared against the classical state of the art represented by simulated annealing (SA) and NSGA-II.

required for decomposition, sampling, and merging, as well as the execution times for all heuristics. For quantum annealing, we utilized D-Wave’s API to measure *QPU Access time*⁶, which includes initialisation, programming, and sampling phases (see Section 3.2.2). We also accounted for the time taken for embedding, a process that, while initially time-consuming, can be reused to achieve substantial speed-ups in subsequent runs.

Table 5. Average time taken to run generate the outputs across different heuristics. Embedding takes up most of time when running quantum annealers, which, in total, still takes less time than simulated annealing and NSGA-II.

	Quantum Annealing			S. Annealing	NSGA-II	Random
	Embedding	Pre and Post	QPU Access			
Suspension	46.61s	<0.2s	3.92s	62.3s	75.2s	<1s
Platooning	41.2s	<0.2s	4.59s	69.1s	78.6s	<1s

Experimental analysis: The results demonstrate noticeable efficiency gain for quantum annealing. In the suspension system case study, quantum annealing required a median of 50.73s total time (of which, only 3.92s were actual QPU access), compared to 62.3s for simulated annealing and 75.2s for NSGA-II. In the platooning system, quantum annealing required 45.99s total (with 4.59s of QPU access), compared to 69.1s and 78.6s for simulated annealing and NSGA-II, respectively. Random selection was the fastest, taking less than one second in both cases but at the cost of significantly lower fault detection rates. Decomposition and re-composition processes for quantum annealing added approximately 0.2s seconds per run. Embedding, on the other hand, took between 40s to 45s, overshadowing QPU access time. Regardless, even with these additional steps, our approach with quantum annealing remained more efficient overall than the alternatives. We also conduct the Mann-Whitney U test and Vargha and Delaney’s $\hat{A}_{12}$ effect size to evaluate the efficiency of our approach with simulated annealing and NSGA-II. We set significance level as 0.05. Results show

⁶https://docs.dwavesys.com/docs/latest/c_qpu_timing.html

that all p-values are smaller than 0.05 with $\hat{A}_{12}$ values smaller than 0.5, demonstrating our approach show significantly higher efficiency than both two other approaches.

5.4.5 Summary of the results. The results of this experiment provide valuable insights into the efficacy and efficiency of quantum annealing for test case generation when compared to state-of-the-art heuristics. In RQ1, we demonstrate that quantum annealing strategies are not ready to solve large scale problems without an accompanying decomposition solution. However, even when considering the time required for decomposition and recomposing the problem and solution, quantum annealing maintained superior overall efficiency. As the main drawback in terms of efficiency, embedding takes a substantial amount of time to complete. This can be mitigated by re-using the same embedding for subsequent runs, making quantum annealing a more scalable solution for larger or iterative problems. More efficient techniques for embedding are still the subject of ongoing research [43, 79]. In terms of effectiveness, we have observed parity in fault detection rates across heuristics, which demonstrates that no approach is statistically superior in this context. This further highlights quantum annealing as a viable alternative to the state-of-the-art heuristics.

In summary, these findings underscore the strengths and trade-offs of quantum annealing. It offers clear advantages in computational efficiency and remains competitive in fault detection, but requires careful management of problem decomposition and embedding to minimise variance and ensure consistent results.

However, in addition to runtime, it is important to acknowledge the cost implications of using quantum annealing. Unlike classical approaches, which can be executed on commodity hardware at negligible additional cost, access to quantum processing units (QPUs) is billed per second of usage. This makes QA substantially more expensive in its current form. For example, in our experiments the improvements achieved must be weighed against the cost of QPU seconds, which may not be justifiable in all contexts. However, in safety-critical CPS scenarios where even small improvements in fault detection or coverage can prevent costly failures, such trade-offs could still be acceptable. We therefore view QA as a complementary technique, where cost-benefit considerations will remain a decisive factor until access models and pricing for quantum hardware evolve.

Moreover, to ensure transparency and fairness, the comparison presented in Section 5.4.3 is intended to illustrate the relative performance of quantum annealing (QA) under practically accessible computational resources, rather than to claim an absolute advantage over classical optimisation methods. The performance of classical algorithms is indeed highly dependent on hardware characteristics, and we have included details on the specifications of the classical machines in Section 5.3.1. Our focus is on demonstrating that QA performs sufficiently well to be considered a viable and competitive optimisation technique for test case generation, even when evaluated under modest conditions.

It is important to note that direct runtime comparisons between quantum and classical systems are inherently imbalanced due to fundamental architectural differences. Quantum processors, such as D-Wave’s annealers, operate at clock frequencies several orders of magnitude lower than those of classical CPUs, making raw time measurements an unfair basis for performance conclusions. Instead, our results should be interpreted as evidence of QA’s capability to achieve comparable or superior solution quality despite these physical limitations. As quantum hardware continues to evolve its relative performance is expected to improve significantly. Therefore, our results are best understood as an early indication of QA’s promise, rather than a definitive claim of superiority over state-of-the-art classical computing systems.

5.5 Threats to Validity

The following threats to validity have been identified for this experiment.

- **Internal Validity:** Our experiment methodology relies on specific types of faults, which might not fully capture the diversity or complexity of faults encountered in real-world CPS applications. In addition, only one fault is inserted at a time to avoid interactions between faults. However, this may not reflect real-world scenarios. To mitigate this issue, we have introduced a total of 50 faults using an external automatic tool found in peer-reviewed literature. As another threat to internal validity, the use of sampling in quantum annealing to handle problem size limitations may introduce bias, as different sampling strategies might yield different results. Investigating the effect of different sampling approaches is outside of the scope of this study and is left for future work.
- **External Validity:** The experiment is conducted on specific Simulink models and, hence, the system under test is a simulation-based model that may lack fidelity. As mitigation to this threat, our models have been developed by domain-experts and have been employed across different experiments available in peer-reviewed literature. As a second threat to external validity, quantum annealing is evaluated under specific problem sizes and sub-problem configurations. Its scalability to even larger problems remains uncertain.
- **Construct Validity:** As a threat to construct validity, the quality of the inputs used in this experiments also have an impact on the results. We mitigate this issue by comparing our approach against three baselines used in test case generation. One that generates random but valid inputs and two search-based approaches; it is unclear whether other forms of input generation would have the same effect.
- **Conclusion Validity:** As a threat to conclusion validity, this experiment only considers two examples of CPS. This makes it hard to generalise the outcome of this experiment for a general class of cyber-physical systems. This is mitigated by the large number of mutants that were inserted into this system. Moreover, the accuracy of the test oracle, which determines fault detection by comparing model outputs to an expected range, may affect the reliability of results. If the oracle parameters for the euclidean distance do not accurately represent acceptable deviations for all faults, this can affect the conclusions. To mitigate these issues, we note that the choice of mutation operators and oracles have been made by experts in the field based on domain knowledge and prior, peer-reviewed, experiments.

6 CONCLUSION

In this work, we introduce a novel approach for efficiently generating effective test cases for Cyber-Physical Systems (CPS) by employing quantum annealing. Our method provides a structured, three-phase process to identify critical regions of test cases based on adequacy metrics, formulate the problem as a combinatorial optimisation using the Quadratic Unconstrained Binary Optimisation (QUBO) model, and apply targeted mutations that enhance fault-detection and test case diversity in the identified regions. Through empirical evaluation on two CPS case studies, we demonstrate that our quantum-based strategy not only improves the effectiveness of a previously generated test suites but also surpasses alternative test case generation methods in terms of efficiency.

Our results affirm that quantum annealing has significant potential to solve complex, optimisation-heavy problems even in test case generation contexts. Although current quantum computing technology has limitations, our findings highlight how it can be adequately utilised within specific problem domains. Future work could explore scaling this approach as quantum hardware and software advance. Moreover, we will explore more sophisticated slicing strategies, such as clustering or heuristic-based partitioning [1], to group points that share similar dynamic or structural properties in the model. This may improve both the efficiency of the annealing process and the relevance of the generated mutations.

ACKNOWLEDGMENTS

Hugo Araujo and Mohammad Reza Mousavi have been partially supported by the UKRI Trustworthy Autonomous Systems Node in Verifiability, Grant Award Reference EP/V026801/2, EP- SRC project on Verified Simulation for Large Quantum Systems (VSL-Q), grant reference EP/Y005244/1 and the EPSRC project on Robust and Reliable Quantum Computing (RoARQ), Investigation 009 Model-based monitoring and calibration of quantum computations (ModeMCQ), grant reference EP/W032635/1 and ITEA/InnovateUK projects GENIUS and GreenCode. Xinyi Wang acknowledges support from Simula's Internal Strategic Project on Quantum Software Engineering. Shaukat Ali is supported by the Norwegian Quantum Software Center (NorQSoft), Project No. 361350, funded by the Research Council of Norway.

REFERENCES

- [1] A. Aerts, B. T. Minh, M. R. Mousavi, and M. A. Reniers, "Temporal logic falsification of cyber-physical systems: An input-signal-space optimization approach," in *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2018, pp. 214–223.
- [2] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, no. 1, p. 015002, 2018.
- [3] R. Alur, *Principles of cyber-physical systems*. MIT Press, 2015.
- [4] H. Araujo, G. Carvalho, M. Mousavi, and A. Sampaio, "Multi-objective search for effective testing of cyber-physical systems," in *Proceedings of the 17th International Conference on Software Engineering and Formal Methods*. Springer, 2019.
- [5] H. Araujo, T. Hoenselaar, M. R. Mousavi, and A. Vinel, "Connected automated driving: A model-based approach to the analysis of basic awareness services," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2020, pp. 1–7.
- [6] A. Arrieta, S. Wang, U. Markiegi, G. Sagardui, and L. Etxeberria, "Search-based test case generation for cyber-physical systems," in *2017 IEEE congress on evolutionary computation (CEC)*. IEEE, 2017, pp. 688–697.
- [7] —, "Employing multi-objective search to enhance reactive test generation and prioritization for testing industrial cyber-physical systems," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018, pp. 7–8.
- [8] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, B. Foxen *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.
- [9] M. Ayodele, "Penalty weights in qubo formulations: Permutation problems," in *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*. Springer, 2022, pp. 159–174.
- [10] E. Bartocci, L. Mariani, D. Ničković, and D. Yadav, "Fim: fault injection and mutation for simulink," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 1716–1720. [Online]. Available: <https://doi.org/10.1145/3540250.3558932>
- [11] A. Bayerstadler, G. Becquin, J. Binder, T. Botter, H. Ehm, T. Ehmer, M. Erdmann, N. Gaus, P. Harbach, M. Hess *et al.*, "Industry quantum computing applications," *EPJ Quantum Technology*, vol. 8, no. 1, p. 25, 2021.
- [12] C. Berghem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa, "Overview of platooning systems," in *Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)*, 2012.
- [13] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical science*, vol. 8, no. 1, pp. 10–15, 1993.
- [14] J. Blank and K. Deb, "pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.
- [15] N. S. Blunt, J. Camps, O. Crawford, R. Izsák, S. Leontica, A. Mirani, A. E. Moylett, S. A. Scivier, C. Sunderhauf, P. Schopf *et al.*, "Perspective on the current state-of-the-art of quantum computing for drug discovery applications," *Journal of Chemical Theory and Computation*, vol. 18, no. 12, pp. 7001–7023, 2022.
- [16] M. Böhme, V.-T. Pham, M.-D. Nguyen, and A. Roychoudhury, "Directed greybox fuzzing," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 2329–2344.
- [17] M. Böhme, V.-T. Pham, and A. Roychoudhury, "Coverage-based greybox fuzzing as markov chain," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1032–1043. [Online]. Available: <https://doi.org/10.1145/2976749.2978428>
- [18] K. Böttinger, P. Godefroid, and R. Singh, "Deep reinforcement fuzzing," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 116–122.
- [19] T. A. Caswell, M. Droettboom, A. Lee, E. S. de Andrade, T. Hoffmann, J. Hunter, J. Klymak, E. Firing, D. Stansby, N. Varoquaux, J. H. Nielsen, B. Root, R. May, P. Elson, J. K. Seppänen, D. Dale, J.-J. Lee, D. McDougall, A. Straw, P. Hobson, hannah, C. Gohlke, T. S. Yu, E. Ma, A. F. Vincent, S. Silvester, C. Moad, N. Kniazev, E. Ernest, and P. Ivanov, "matplotlib/matplotlib: Rel: v3.4.3," Aug. 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.5194481>
- [20] S. K. Cha, T. Avgerinos, A. Rebert, and D. Brumley, "Unleashing mayhem on binary code," in *2012 IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 380–394.
- [21] C. Chen, B. Cui, J. Ma, R. Wu, J. Guo, and W. Liu, "A systematic review of fuzzing techniques," *Computers & Security*, vol. 75, pp. 118–137, 2018.

- [22] V. Chipounov, V. Kuznetsov, and G. Candea, "S2e: A platform for in-vivo multi-path analysis of software systems," *Acm Sigplan Notices*, vol. 46, no. 3, pp. 265–278, 2011.
- [23] B. A. Cipra, "An introduction to the ising model," *The American Mathematical Monthly*, vol. 94, no. 10, pp. 937–959, 1987.
- [24] E. Cohen and B. Tamir, "D-wave and predecessors: From simulated to quantum annealing," *International Journal of Quantum Information*, vol. 12, no. 03, p. 1430002, 2014.
- [25] D-Wave, "D-wave documentation," <https://docs.dwavequantum.com/en/latest/>, [Online; accessed 16-October-2025].
- [26] —, "Reformulating a problem," https://docs.dwavequantum.com/en/latest/quantum_research/reformulating.html, [Online; accessed 27-October-2025].
- [27] B. De Schutter, W. Heemels, J. Lunze, C. Prieur *et al.*, "Survey of modeling, analysis, and control of hybrid systems," *Handbook of Hybrid Systems Control—Theory, Tools, Applications*, pp. 31–55, 2009.
- [28] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [29] J. Deshmukh, M. Horvat, X. Jin, R. Majumdar, and V. S. Prabhu, "Testing cyber-physical systems through bayesian optimization," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, pp. 1–18, 2017.
- [30] S. Documentation, "Simulation and model-based design," 2020. [Online]. Available: <https://www.mathworks.com/products/simulink.html>
- [31] A. Donzé, "Breach, a toolbox for verification and parameter synthesis of hybrid systems," in *Computer Aided Verification: 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings 22*. Springer, 2010, p. 167–170.
- [32] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, "Quantum computing for finance: State-of-the-art and future prospects," *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–24, 2020.
- [33] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum computation by adiabatic evolution," *arXiv preprint quant-ph/0001106*, 2000.
- [34] D. Fortunato, J. Campos, and R. Abreu, "Mutation testing of quantum programs: A case study with qiskit," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–17, 2022.
- [35] —, "Qmutpy: A mutation testing tool for quantum algorithms and applications in qiskit," in *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2022, pp. 797–800.
- [36] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.
- [37] M. Hamzah, M. M. Islam, S. Hassan, M. N. Akhtar, M. J. Ferdous, M. B. Jasser, and A. W. Mohamed, "Distributed control of cyber physical system on various domains: A critical review," *Systems*, vol. 11, no. 4, p. 208, 2023.
- [38] H. Hussein, A. Younes, and W. Abdelmoez, "Quantum algorithm for solving the test suite minimization problem," *Cogent Engineering*, vol. 8, no. 1, p. 1882116, 2021.
- [39] D.-W. S. Inc., "Reformulating a problem for the d-wave system (native qpu formulations: Ising and qubo)," Online documentation, 2024, accessed: November 2025. [Online]. Available: https://docs.dwavequantum.com/en/latest/quantum_research/reformulating.html
- [40] K. Jun and H. Lee, "Hubo and qubo models for prime factorization," *Scientific Reports*, vol. 13, no. 1, p. 10080, 2023.
- [41] F. M. Kifetew, R. Tiella, and P. Tonella, "Generating valid grammar-based test inputs by means of genetic programming and annotated grammars," *Empirical Software Engineering*, vol. 22, no. 2, pp. 928–961, 2017.
- [42] J. Kim, S. Chon, and J. Park, "Suggestion of testing method for industrial level cyber-physical system in complex environment," in *2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2019, pp. 148–152.
- [43] M. S. Könz, W. Lechner, H. G. Katzgraber, and M. Troyer, "Embedding overhead scaling of optimization problems in quantum annealing," *PRX Quantum*, vol. 2, no. 4, p. 040322, 2021.
- [44] E. A. Lee, "Cyber physical systems: Design challenges," *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, pp. 363–369, 2008.
- [45] F. Leymann and J. Barzen, "The bitter truth about gate-based quantum algorithms in the nisq era," *Quantum Science and Technology*, vol. 5, no. 4, p. 044007, 2020.
- [46] N. Lyamin, A. Vinel, M. Jonsson, and B. Bellalta, "Cooperative awareness in VANETs: On ETSI EN 302 637-2 performance," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 17–28, 2017.
- [47] A. Mateen, M. Nazir, and S. A. Awan, "Optimization of test case generation using genetic algorithm (ga)," *arXiv preprint arXiv:1612.08813*, 2016.
- [48] R. Matinejad, S. Nejati, L. Briand, T. Bruckmann, and C. Poull, "Search-based automated testing of continuous controllers: Framework, tool support, and case studies," *Information and Software Technology*, vol. 57, pp. 705–722, 2015.
- [49] R. Matinejad, S. Nejati, L. C. Briand, and T. Bruckmann, "Effective test suites for mixed discrete-continuous stateflow controllers," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, 2015, pp. 84–95.
- [50] E. Mendiluze, S. Ali, P. Arcaini, and T. Yue, "Muskit: A mutation analysis tool for quantum software testing," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2021, pp. 1266–1270.
- [51] C. Menghi, S. Nejati, L. Briand, and Y. I. Parache, "Approximation-refinement testing of compute-intensive cyber-physical models: An approach based on system identification," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 372–384.
- [52] C. Miller, Z. N. Peterson *et al.*, "Analysis of mutation and generation-based fuzzing," *Independent Security Evaluators, Tech. Rep.*, vol. 4, 2007.

- [53] A. Miransky, "Using quantum computers to speed up dynamic testing of software," in *Proceedings of the 1st International Workshop on Quantum Programming for Software Engineering*, 2022, pp. 26–31.
- [54] A. Montanaro, "Quantum algorithms: An overview," *npj Quantum Information*, vol. 2, p. 15023, 2016.
- [55] P. J. Mosterman and J. Zander, "Cyber-physical systems challenges: a needs analysis for collaborating embedded software systems," *Software & Systems Modeling*, vol. 15, no. 1, pp. 5–16, 2016.
- [56] M. Motta and J. E. Rice, "Emerging quantum computing algorithms for quantum chemistry," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 12, no. 3, p. e1580, 2022.
- [57] O. Müller and T. Stauner, "Modelling and verification using linear hybrid automata—a case study," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 6, no. 1, pp. 71–89, 2000.
- [58] N. Nichols, M. Raugas, R. Jasper, and N. Hilliard, "Faster fuzzing: Reinitialization with deep neural models," *arXiv preprint arXiv:1711.02807*, 2017.
- [59] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.
- [60] P. Niroula, R. Shaydulin, R. Yalovetzky, P. Minssen, D. Herman, S. Hu, and M. Pistoia, "Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer," *Scientific Reports*, vol. 12, no. 1, p. 17171, 2022.
- [61] J. Offutt, S. Liu, A. Abdurazik, and P. Ammann, "Test oracle strategies for model-based testing," in *Proceedings of the International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 2008, pp. 136–145.
- [62] A. Panichella, F. M. Kifetew, and P. Tonella, "Automated test case generation as a many-objective optimisation problem with dynamic selection of the targets," *IEEE Transactions on Software Engineering*, vol. 44, no. 2, pp. 122–158, 2018.
- [63] E. Pelofske, A. Bärtschi, and S. Eidenbenz, "Qaoa: 127 qubit higher-order ising problems on nisq computers," in *International Conference on High Performance Computing*. Springer, 2023, pp. 240–258.
- [64] A. Perdomo-Ortiz, J. Fluegemann, S. Narasimhan, R. Biswas, and V. N. Smelyanskiy, "A quantum annealing approach for fault detection and diagnosis of graph-based systems," *The European Physical Journal Special Topics*, vol. 224, pp. 131–148, 2015.
- [65] V.-T. Pham, M. Böhme, and A. Roychoudhury, "Model-based whitebox fuzzing for program binaries," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 543–553. [Online]. Available: <https://doi.org/10.1145/2970276.2970316>
- [66] A. Pretschner and J. Philipps, "Model-based testing," *Software and Systems Modeling*, vol. 4, no. 2, pp. 105–108, 2005.
- [67] A. P. Punnen, "The quadratic unconstrained binary optimization problem," *Springer International Publishing*, vol. 10, pp. 978–3, 2022.
- [68] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," *Design Automation Conference*, pp. 731–736, 2010.
- [69] Z. Sadri-Moshkenani, J. Bradley, and G. Rothermel, "Survey on test case generation, selection and prioritization for cyber-physical systems," *Software Testing, Verification and Reliability*, vol. 32, no. 1, p. e1794, 2022.
- [70] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.
- [71] R. A. Silva, S. d. R. S. de Souza, and P. S. L. de Souza, "A systematic review on search based mutation testing," *Information and Software Technology*, vol. 81, pp. 19–35, 2017.
- [72] Student, "The probable error of a mean," *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908.
- [73] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.
- [74] X. Wang, S. Ali, A. Arrieta, P. Arcaini, and M. Arratibel, "Application of quantum extreme learning machines for qos prediction of elevators' software in an industrial context," in *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, ser. FSE 2024. New York, NY, USA: Association for Computing Machinery, 2024, p. 399–410. [Online]. Available: <https://doi.org/10.1145/3663529.3663859>
- [75] X. Wang, S. Ali, T. Yue, and P. Arcaini, "Quantum approximate optimization algorithm for test case optimization," *IEEE Transactions on Software Engineering*, 2024.
- [76] X. Wang, A. Muqet, T. Yue, S. Ali, and P. Arcaini, "Test case minimization with quantum annealers," *ACM Trans. Softw. Eng. Methodol.*, Jul. 2024, just Accepted. [Online]. Available: <https://doi.org/10.1145/3680467>
- [77] X. Wang, T. Yu, P. Arcaini, T. Yue, and S. Ali, "Mutation-based test generation for quantum programs with multi-objective search," in *Proceedings of the genetic and evolutionary computation conference*, 2022, pp. 1345–1353.
- [78] J. Yang, S. Arya, and Y. Wang, "Formal-guided fuzz testing: Targeting security assurance from specification to implementation for 5g and beyond," *IEEE Access*, 2024.
- [79] S. Zbinden, A. Bärtschi, H. Djidjev, and S. Eidenbenz, "Embedding algorithms for quantum annealers with chimera and pegasus connection topologies," in *International Conference on High Performance Computing*. Springer, 2020, pp. 187–206.
- [80] M. Zhang, S. Ali, and T. Yue, "Uncertainty-wise test case generation and minimization for cyber-physical systems," *Journal of Systems and Software*, vol. 153, pp. 1–21, 2019.