SOLQHEALER: QUANTUM PROCEDURES FOR RENDERING INFEASIBLE SOLUTIONS FEASIBLE: A PROOF OF CONCEPT WITH THE MAXIMUM INDEPENDENT SET PROBLEM AND 3-SAT

Samuel Deleplanque, Luis Fernando Pérez Armas, Stefan Creemers

REPRINT | 3337

UCLouvain

CORE | LIDAM
Louvain Institute of Data Analysis and Modeling in economics and statistics

`solQHealer`: Quantum Procedures for Rendering Infeasible Solutions Feasible: A Proof of Concept with the Maximum Independent Set Problem and 3-SAT

Samuel DELEPLANQUE[1*], Luis Fernando PÉREZ ARMAS[2] and Stefan CREEMERS[3]

[1*]Univ. Lille, CNRS, Centrale Lille, Junia, Univ. Polytechnique Hauts-de-France, UMR 8520 - IEMN, 41 Bd Vauban, Lille, 59000, Nord, France.
[2]Operations Management, IESEG School of Management, Univ. Lille, CNRS, UMR 9221 - LEM - Lille Economie Management, 3 rue de la digue, Lille, F-59800, Nord, France.
[3]Center for Operations Research and Econometrics (CORE), Université catholique de Louvain, Voie du Roman Pays 34, Louvain-la-Neuve, 1348, Belgium.

*Corresponding author(s). E-mail(s): samuel.deleplanque@junia.com;
Contributing authors: l.perezarmas@ieseg.fr; stefan.creemers@uclouvain.be;

Abstract

Over the past decade, the usefulness of quantum annealing hardware for combinatorial optimization has been the subject of active debate. Although current analog quantum machines do not guarantee optimality, operating instead as heuristic solvers, the technology is evolving rapidly. Beyond performance alone, this emerging technologies offers fundamentally new approaches to problem-solving that are not readily accessible to classical exact methods particularly in dynamic environments or online optimization settings. This paper focuses on one such approaches: Reverse Quantum Annealing (RQA). Unlike classical exact methods, RQA allows the optimization process to begin from an initial infeasible solution by embedding it directly into the qubits' initial state. We leverage this capability by formulating problem constraints as penalty terms within Quadratic Unconstrained Binary Optimization (QUBO) models, thereby preserving infeasible solutions within the quantum search space. We propose iterative strategies that apply RQA in three distinct modes to rapidly repair infeasible solutions. These methods are evaluated on two well-known NP-hard problems: the Maximum Independent Set (MIS) and the 3-SAT problem. Our results demonstrate the effectiveness of RQA in steering infeasible configurations toward feasibility, offering promising potential for real-time applications where problem data may change suddenly and solutions must be repaired swiftly.

Keywords: Reverse Quantum Annealing, Quantum optimization, Maximum Independent Set, 3-SAT

1 Introduction

Reverse Quantum Annealing (RQA) is a process derived from Quantum Annealing (QA) that can be implemented on analog quantum machines such as the D-Wave system. Unlike traditional QA, which starts from a random initial state, RQA begins with a known solution that one aims to improve. To perform RQA, the quantum machine requires this initial, known solution and a Quadratic Unconstrained Binary Optimization problem (QUBO) formulation of the problem. The QUBO is then transformed into an Ising model (isomorph of the QUBO formulation but with -1 and +1 variable values) that is solved by the quantum machine. QA has been shown theoretically to have superior performance when compared to Simulated Annealing (SA) [1]. Also experimentally, a speedup has been demonstrated [2]. This study explores the application of RQA in a specific context where an infeasible solution is available and is used as an initial solution in order to find a feasible solution as fast as possible.

Consider solving a problem $\mathcal{P}$ (e.g., rescheduling of trains and any other problem for which online optimization is required) for which we have an infeasible solution $S^{\mathcal{P}}_{\text{inf}}$. This solution may have interesting characteristics but cannot be used due to one or more constraints that are violated. Using the QUBO formulation of the problem, an analog quantum machine can take this solution $S^{\mathcal{P}}_{\text{inf}}$ as input, similar to a classical solver that takes a Mixed Integer Linear Program (MILP) model and a feasible solution as input. The key difference here is that, in the classical case, the solution must be feasible and typically serves to establish a bound for pruning a Branch-and-Bound tree. In our case, an infeasible solution (i.e., $S^{\mathcal{P}}_{\text{inf}}$) serves as the starting point for RQA exploration, which will guide the evolution towards a feasible solution (and even an optimal solution if allowed enough time).

RQA and QA are metaheuristics. Our primary focus here is to make an infeasible solution feasible. The goal is not necessarily to reach the optimal solution. By weighing the penalties in the objective function related to constraint relaxation, we can prioritize feasibility during RQA exploration. Unlike classical solvers, the quantum machine that solves a QUBO does not consider constraints (as the QUBO is an unconstrained optimization problem), thus any variable assignment forms a feasible solution. Rather than working with constraints, we work with penalties to ensure that the exploration eventually yields a feasible solution for the original constrained problem $\mathcal{P}$.

The exploration of the solution space by QA and RQA is very different from classical methods. If we have an infeasible solution, a QUBO formulation allows us to treat this solution as feasible (since it is unconstrained). Although RQA is based on QA, its performance is difficult to evaluate. In the case of highly reactive systems, where the time to come up with a (good) solution is limited, a quantum method like RQA might be suitable. Indeed, the brief duration of the RQA process (just a few microseconds) could be suitable for a problems that only allow a limited time to find a solution (as long as the embedding process in the machine's qubit graph is not too time-consuming). While using optimization for solving real-life problems, unexpected changes can add new constraints that may render in-place solutions infeasible, this can also manifest itself in applications of online optimization [3]. For instance, train systems are highly disrupted, being prone to domino effects, and their rescheduling requires obtaining a solution within a limited timeframe, based on currently available information. Recalculating a new solution with these additional constraints can sometimes take considerable time. With RQA, updating the QUBO can be done quickly by relaxing the new constraints in the objective function.

This work stems from a simple idea that, to the best of our knowledge, has not yet been explored. We propose a general procedure called `solQHealer`, applicable to any problem that uses RQA, to make an infeasible solution feasible in a limited amount of time; either if the solution is no longer feasible (in the case of changes to the problem definition) or if the solution never was feasible (for instance, when no feasible solution was found by a classical computer). `solQHealer` can be used as a subroutine in both quantum and classical procedures. As input, `solQHealer` uses a QUBO model and a a solution that is infeasible with respect to the original problem. This solution in then used as the starting point of the RQA exploration. With carefully adjusted weights for the multipliers that penalize the relaxed constraints, we can guide the evolution of the solution towards feasibility. To validate this procedure, we use RQA to find feasible solutions for the Maximum Independent Set (MIS) problem, a suitable candidate problem for a QUBO reformulation. In addition, we present `solQHealer2`, a method for finding feasible solutions to a decision problem (3-SAT) starting from an infeasible one. In our experiments, we selected instances that can be solved on today's analog quantum machines, as the goal here is not to assess their performance but to demonstrate their use in finding a feasible solution. We assess the similarity between the infeasible solution and the one obtained by computing their cosine similarity.

Because the hardware of current analog quantum machines cannot solve medium or large instances, in this paper, we are limited to solving instances that pose no difficulty for classical computers. Nevertheless, it is important to already develop procedures that can be used for solving solving large instances once quantum computers with a sufficient number of qubits become available.

This paper is organized as follows. Section 2 provides a review of the current literature on quantum annealing and reverse quantum annealing. In Section 3, we formally introduce the heuristic procedure `solQHealer`.

Sections 4 and 5 describe the application of `solQHealer` and `solQHealer2` to the Maximum Independent Set and 3-SAT problems, respectively. Section 6 outlines the experimental setup, instance selection, benchmarking methodology, and presents the computational results. Finally, Section 7 discusses the limitations of our findings and highlights future research opportunities for `solQHealer`. Additional analyses and supplementary information are provided in the appendices.

2 Quantum Annealing and Reverse Quantum Annealing

In this section, we focus on the core principles of adiabatic evolution, quantum annealing, and reverse quantum annealing. We do not aim to provide a foundational introduction to basic quantum mechanics concepts such as qubits, spin systems, superposition, entanglement, and measurement. Readers seeking a thorough grounding in these fundamental topics are encouraged to consult the detailed expositions in [4], [5], and [6].

2.1 Quantum Annealing

QA shares many similarities with its classical counterpart, SA [7]. The SA algorithm involves creating a discrete-time Markov chain operating within the feasible solution space $\mathcal{S}$. Ideally, the transition probabilities of this Markov chain must be designed to ensure that its limiting distribution converges toward the optimal solution of an optimization problem $\mathcal{P}$.

Similarly, the QA algorithm mirrors SA and can be introduced by considering a continuous-time stochastic process operating on the solution space $\mathcal{S}$ (or on a relaxation of $\mathcal{S}$) with an Energy objective function of discrete Variables $s_i \in \{-1, 1\} \forall\, i \in \mathcal{Y}$. More precisely, following [1] and [8], the QA algorithm constitutes a stochastic algorithm for minimizing a quadratic expression, traditionally known as Ising models of the form:

$$E(\sigma) = \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} J_{i,j} s_i s_j + \sum_{i \in \mathcal{S}} h_i s_i$$

Where the parameters, $J_{i,j}$ and h_i define the quadratic and linear coefficients of this function, respectively. Equivalently the same model can also be expressed with a binary solution space $\mathcal{X}$, with objective of minimizing the quadratic expression $\boldsymbol{x}^\top Q \boldsymbol{x}$, known as a QUBO, where decision variables are required to take binary values $\boldsymbol{x} \in \{0, 1\}^n$, where n corresponds to the number of decision variables of problem $\mathcal{P}$. Note that the transformation between the Ising and QUBO problems can be obtained using:

$$x_i = \frac{1 + s_i}{2}.$$

The goal of the QA procedure is to construct then a probability distribution over binary vectors in $\{0, 1\}^n$. This distribution defines the continuous-time stochastic process $\{|\Psi(t)|^2 : t \geq 0\}$, whose dynamics are governed by the Schrödinger differential equation:

$$i\hbar \frac{\partial \Psi(t)}{\partial t} = \mathcal{H}(t, \boldsymbol{x}) \Psi(t),$$

where i is the imaginary constant, $\hbar$ is an exogenous quantity (the reduced Planck constant), and $\mathcal{H}(t, \boldsymbol{x}) : \mathbb{R} \times \{0, 1\}^{2^n} \to \mathbb{R}$ is referred to as the Hamiltonian.

The dynamics of this time-dependent Hamiltonian function are theoretically governed by the universal model of adiabatic evolution [9]. The adiabatic theorem establishes that if a quantum system is subjected to a slow and continuous change in its Hamiltonian, then the system will remain in its instantaneous eigenstate during the change [1, 8].

The QA algorithm is physically implemented using analog control devices to manipulate a collection of qubit states (i.e., quantum bits of information) following a time-dependent Hamiltonian represented as:

$$\mathcal{H}(t, x) = A(t)\mathcal{H}_0(x) + B(t)\mathcal{H}_1(x),$$

where $A(t)$ is a monotonically increasing function of time and $B(t)$ is a monotonically decreasing function of time. The combination of these two functions is known as the annealing schedule, and a typical anneal schedule is represented in Figure 1.

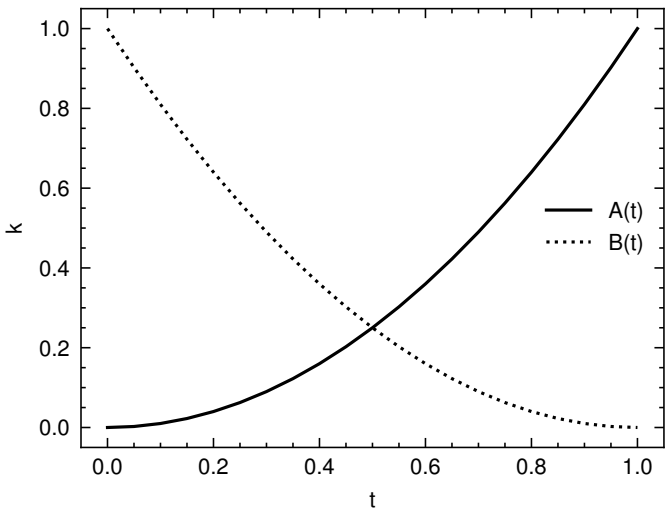


Fig. 1: Example annealing schedule. The y-axis shows the anneal fraction k that represents the overall proportion of how much the system has evolved into $\mathcal{H}_0$. The x-axis represents the elapsed time.

The QA algorithm orchestrates a gradual transition from the initial ground state of an initial Hamiltonian $\mathcal{H}_0$, known as the "*tunneling*" or "*Mixing*" Hamiltonian, to a state described by the problem Hamiltonian $\mathcal{H}_1$. The Hamiltonian $\mathcal{H}_1$ mirrors the energy function of the optimization problem $\mathcal{P}$, ensuring that the ground state for $\mathcal{H}_1$ corresponds to a minimum-cost solution to the optimization problem $\mathcal{P}$. Typically, the problem Hamiltonian $\mathcal{H}_1$ is implemented using the Ising model.

The typical tunneling Hamiltonian $\mathcal{H}_0$ used in QA is defined as:

$$\mathcal{H}_0 = \sum_i \sigma_x x_i,$$

where σ_x represents the Pauli-X operator that is applied to qubit i and that is defined as:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The choice of this particular Hamiltonian as $\mathcal{H}_0$ is motivated by the fact that its ground state has been extensively studied in the literature and can be efficiently computed in polynomial time [4, 10]. The ground state of the tunneling Hamiltonian $\mathcal{H}_0$ is the so-called "*uniform superposition*" state, in which each qubit is equally likely to be in the state $x_i = 0$ or $x_i = 1$, with 50% probability for each. This initial state represents a maximum entropy configuration, effectively corresponding to sampling each binary variable as if flipping a fair coin, ensuring all possible solutions are equally probable at the start of the evolution.

The advantage of solving optimization problems using the continuous-time stochastic process of QA lies in the unique properties of quantum mechanics (i.e. quantum superposition, entanglement, etc.). Unlike classical simulation methods, the QA algorithm allows a system of quantum particles to naturally evolve over time according to the Schrödinger equation, without making measurements; without the computational need to run multiple simulations to attain an empirical approximation of the limit distribution (as would be the case with SA). Note that QA is a general optimization framework capable of solving numerous optimization problems [5, 11, 12], as the ISING/QUBO models are both NP-hard and NP-complete, as demonstrated by [13].

2.2 Reverse Quantum Annealing

Traditional QA starts from the ground state of the tunneling Hamiltonian $\mathcal{H}_0$ (i.e., from an initial superposition where each qubit is put in a uniform superposition state) and evolves towards the target problem Hamiltonian $\mathcal{H}_1$. However, a more sophisticated evolution approach can also be applied. In RQA, instead of starting the annealing process from the ground state of $\mathcal{H}_0$, the process starts from an already available solution $\mathcal{S}_0$. RQA reintroduces partial segments of the tunneling Hamiltonian $\mathcal{H}_0$ for a given time period t' by operating in reverse, thereby reinstating a partial quantum superposition, to then proceed to once again evolve forward into the problem Hamiltonian $\mathcal{H}_1$. This unique reverse and forward progression is used to iteratively refine the initial solution $\mathcal{S}_0$ [14–17], with the expectation of faster convergence into the ground state of the problem Hamiltonian $\mathcal{H}_1$ (i.e., the optimal solution of problem $\mathcal{P}$).

RQA is similar to the concept of reheating in SA. Departing from a warm start, the RQA algorithm performs a neighborhood search, applying superposition to some of its variables, and then continues with the traditional forward annealing evolution. The amount of superposition in the reverse process can be controlled by the user by setting up the amount of time during which the reverse process will take place. The more time t' give to the reverse schedule, the more the annealing process will approach a full uniform superposition state (ground state of the tunneling Hamiltonian $\mathcal{H}_0$), very much similar to a process that increases the temperature in SA. Figure 2 shows a typical RQA schedule.

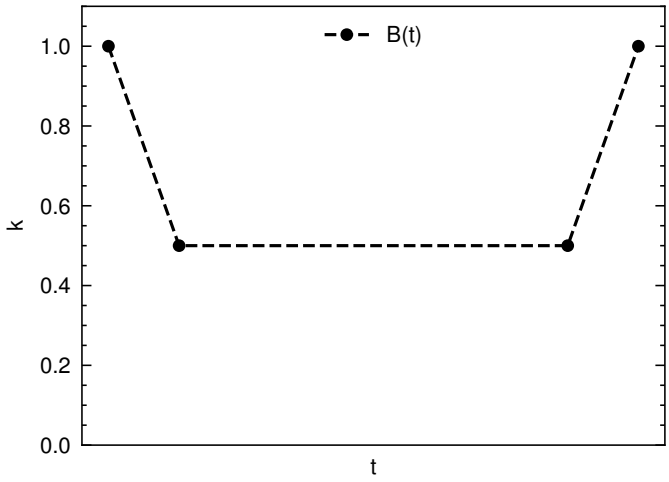


Fig. 2: Reverse annealing schedule. The y-axis shows the anneal fraction k that represents the overall proportion of how much the system has evolved into $\mathcal{H}_0$. The x-axis represents the elapsed time.

The forward annealing process of RQA still seeks a final state of minimum energy but departs from a warm start solution $\mathcal{S}_0$ rather than a uniform superposition of all qubits (maximum entropy state). The rationale behind this approach is that a feasible solution is likely to be close to another feasible solution, as opposed to being anywhere within the state of all possible solutions (which is assumed when using a uniform superposition).

For the reasons discussed in the Appendix B, and in line with the findings of [1], we argue that in the context of repairing infeasible solutions, the RQA-based procedure introduced in Section 3 offers distinct advantages over traditional simulated annealing.

First, RQA algorithms can distinguish infeasible solutions with greater precision [11]. These solutions can be heavily penalized without requiring excessively high temperatures, thus avoiding the risk of turning the algorithm into a random sampler. Moreover, as demonstrated by [14, 18], quantum annealing and reverse quantum annealing can leverage quantum phenomena such as quantum tunneling to effectively escape neighborhoods of local minima.

Second, and perhaps more importantly, RQA explores the neighborhood of an infeasible solution more thoroughly than SA. This is because the application of the *tunneling* Hamiltonian $\mathcal{H}_0$ is centered around the initial warm-start solution $\mathcal{S}_0$, even if that solution is infeasible.

3 *solQHealer*

Let $S^{\mathcal{P}}_{\text{inf}}$ be an infeasible solution of a problem $\mathcal{P}$ that we wish to make feasible, and $f^{\mathcal{P}}_{\text{QUBO}}$ the unconstrained binary model of $\mathcal{P}$ where all constraints have been transformed into penalties. These two elements form the input to RQA. The Unified Modeling Language (UML) diagram in Figure 3 represents an overview of the process that seeks to make $S^{\mathcal{P}}_{\text{inf}}$ feasible. In the diagram, the initial node is represented by a black circle and marks the starting point of the activity. Conversely, the final node is represented by a black circle with a border and signifies the end of the activity. Activities within the workflow are represented as rounded rectangles, each corresponding to a distinct task or action.

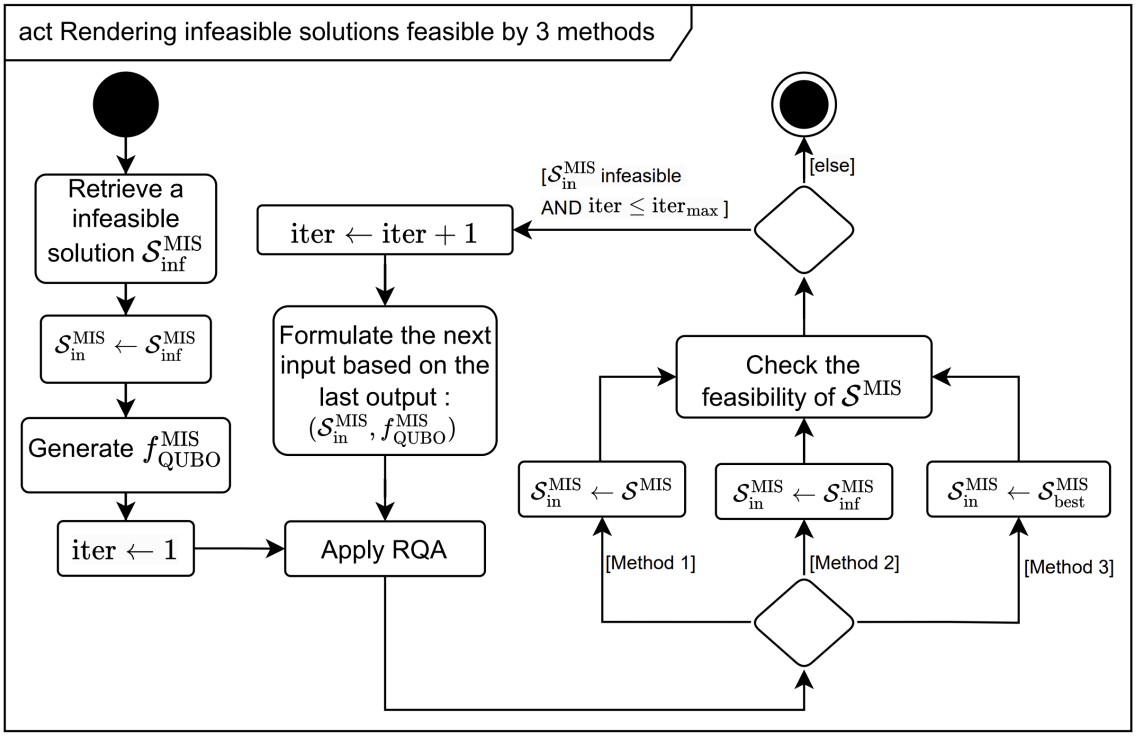


Fig. 4: UML activiy diagram: Making infeasible solutions feasible methods.

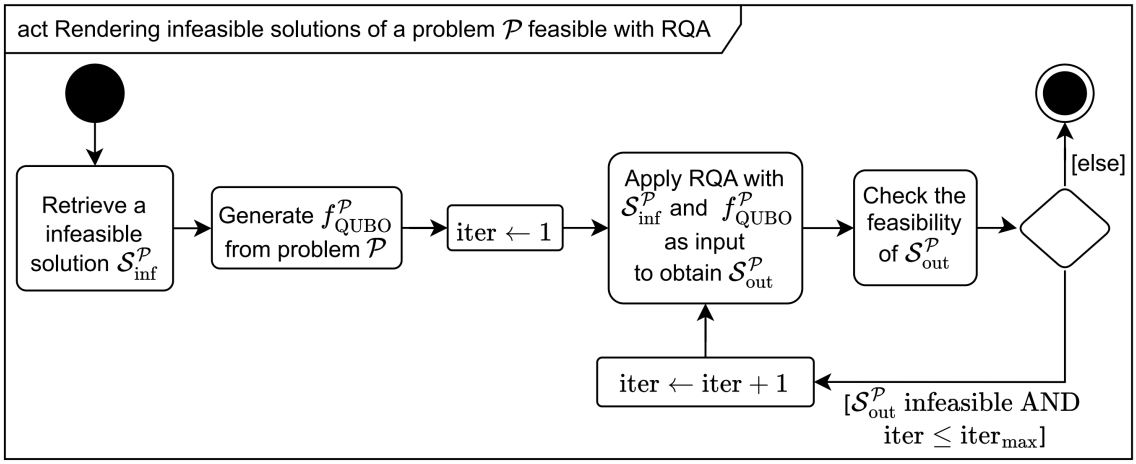


Fig. 3: UML activity diagram: `solQHealer` process to render infeasible solutions of a problem $\mathcal{P}$ feasible.

The main idea is to propose a generic procedure for applying RQA to obtain a feasible solution when a QUBO model and an infeasible solution are provided as inputs. Similar to QA, which iterates annealings, RQA also iterates explorations. Figure 3 illustrates an iterative procedure that uses RQA until a feasible solution is found or until the maximum number of iterations ($\text{iter}_{\max}$) has been reached. To have control over what is done between each iteration, we propose three straightforward methods based on RQA. As illustrated in the UML diagram in Figure 4, after retrieving an infeasible solution and generating the QUBO for problem $\mathcal{P}$, these two elements constitute the input for the first iteration of the RQA. The feasibility of the obtained solution is then tested, and the process is iterated according to one of three methods that operate on the different outputs of the RQA or the initial infeasible solution. Each method treats the subsequent infeasible solution as part of the input in a distinct manner:

- *Method 1:* This method takes the last obtained solution $\mathcal{S}^{\mathcal{P}}_{\text{out}}$ as the input for the RQA.
- *Method 2:* This method consistently uses the first infeasible solution $\mathcal{S}^{\mathcal{P}}_{\text{inf}}$ obtained as the input for the RQA.
- *Method 3:* This method utilizes the current best obtained solution $\mathcal{S}^{\mathcal{P}}_{\text{best}}$ as the input for the RQA.

Here, $\mathcal{S}^{\mathcal{P}}_{\text{best}}$ corresponds to the solution that has the smallest relative distance from a feasible solution. In an infeasible solution, the number of non-zero penalties in the QUBO corresponds to the number of violated constraints. For some problems (such as MIS), these can easily be identified by the high values assigned to the multipliers, denoted as λ, which have been selected such that the violation of each constraint corresponds to a value greater than a maximum limit that is set on the number of vertices of the independent set. Therefore, we define a metric of distance towards feasibility by aggregating the count of violated constraints. Subsequently, $\mathcal{S}^{\mathcal{P}}_{\text{best}}$ is identified as the solution that currently minimizes the most the number of such violations.

4 Application to the Maximum Independent Set Problem

The maximum independent set is an NP-hard problem as noted by [19, 20] that uses a simple undirected graph $\mathcal{G}$ composed of vertices $\mathcal{V}$ and edges $\mathcal{E}$. An optimal solution to the MIS corresponds to the largest subset of vertices $\mathcal{S} \subseteq \mathcal{V}$ where each pair of vertices is non-adjacent. This problem is discussed in depth by[21] and further explored in [22].

When using linear programming to solve MIS, we define binary decision variables x_k for each $k : k \in \mathcal{V}$:

$$x_k = \begin{cases} 1 \text{ if vertex } k \text{ is in the independant set,} \\ 0 \text{ otherwise.} \end{cases} \tag{1}$$

The objective function $f^{\text{MIS}}(x)$, aims to maximize the size of $\mathcal{S}$, and can be expressed as:

$$f^{\text{MIS}}(x) = \sum_{k \in \mathcal{V}} x_k. \tag{2}$$

The sole constraint, for the MIS requires that for every edge (i, j) within $\mathcal{E}$ no two adjacent vertices can be selected, and is given by the expression:

$$x_i + x_j \leq 1 \quad \forall (i, j) \in \mathcal{E}. \tag{3}$$

Constraint 3 can be reformulated to be compatible with quantum machines that support QUBO or Ising models. Specifically, it can be expressed as a quadratic function:

$$x_i x_j = 0 \quad \forall (i, j) \in \mathcal{E}. \tag{4}$$

We introduce λ as the penalty multiplier associated with constraint (4) in the QUBO formulation. The resulting QUBO objective function, denoted by $f^{\text{MIS}}_{\text{QUBO}}(x)$, for solving the Maximum Independent Set problem can then be written as:

$$f^{\text{MIS}}_{\text{QUBO}}(x) = -\sum_{k \in \mathcal{V}} x_k + \lambda \sum_{(i,j) \in \mathcal{E}} x_i x_j. \tag{5}$$

It remains a challenge to assess which aspect of quantum hardware provides greater precision in analog quantum machines: the mechanism responsible for assigning biases to individual variables (linear terms), or the one used for assigning weights to quadratic interactions between variable pairs. In general, QUBO problems are composed of two main components: one representing the original objective function and another encoding constraint violations through weighted penalty terms. If the hardware responsible for generating linear biases (i.e., the weights assigned to individual qubits) is believed to be more accurate than the hardware handling pairwise interactions (i.e., the weights of quadratic terms), it may be preferable to formulate the QUBO problem in a way that emphasizes linear terms. In such cases, one might consider solving the following QUBO formulation:

$$f'^{\text{MIS}}_{\text{QUBO}}(x) = -\lambda' \sum_{k \in \mathcal{V}} x_k + \sum_{(i,j) \in \mathcal{E}} x_i x_j. \tag{6}$$

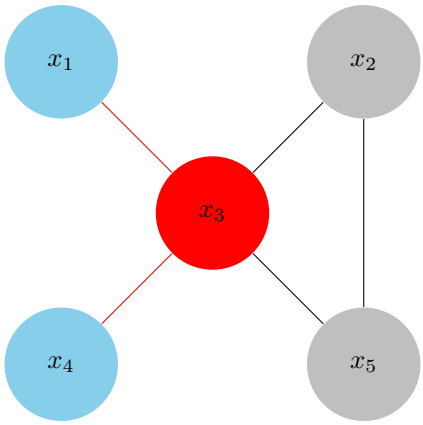
Although still limited, the body of literature on QUBO formulations has grown steadily in recent years [23–25]. This increase parallels the emergence of quantum computing platforms that accept QUBO models either directly as input such as in the case of D-Wave or Pasqal devices [26, 27] or indirectly through algorithms that require QUBO based representations, such as the Quantum Approximate Optimization Algorithm (QAOA) [10].

The values of the penalty multipliers λ, which determine the weight assigned to constraint violations, are typically calibrated to guide the search toward the desired optimal solution. It is important to note that, in some cases, certain infeasible solutions may have objective function values similar to those of feasible ones. To avoid this ambiguity during the process of transforming an infeasible solution into a feasible one, we ensure

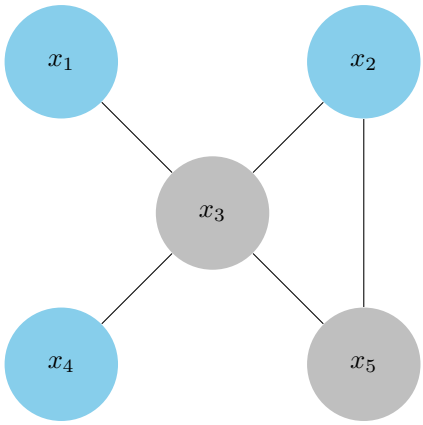
that the objective values of infeasible solutions are clearly distinguishable from those of feasible solutions. To achieve this separation, the penalty multiplier is set equal to the number of vertices, that is, $\lambda = |\mathcal{V}|$.

Example. Consider a graph $\mathcal{G}$ with five vertices as depicted in Figure 5. After applying an initial procedure for solving the MIS problem, infeasible solution $S_{\text{inf}}^{\text{MIS}}(x)$ is obtained, as shown in Figure 5a. Including vertex x_3 in the solution along with x_1 and x_4 incurs an additional cost to the QUBO of 2λ, since $x_1x_3 = x_3x_4 = 1$. The value assigned to λ is set such that this added cost to the QUBO exceeds the absolute value of any additional (negative) value that the selection of any set of vertices as independent might have contributed. As $\lambda = |\mathcal{V}|$, the impact of incurring λ even once cannot be countered by any selection of set of vertices (i.e., the objective function is always greater-than-or-equal-to 0). The gap between the optimal solution and $S_{\text{inf}}^{\text{MIS}}(x)$ can be viewed as the energy difference between the ground state (optimal) and the energy of the excited state corresponding to $S_{\text{inf}}^{\text{MIS}}(x)$ of an adiabatic process.

RQA takes as parameters $f_{\text{QUBO}}^{\text{MIS}}(x)$ and the solution $S_{\text{inf}}^{\text{MIS}}(x)$, which is infeasible for the problem but feasible for the associated QUBO (albeit with a potentially high objective value due to the penalties that are incurred due to violating constraints). The metaheuristic explores the solution space (thus always feasible) in a manner that reduces energy until a negative value is obtained. If the ground state is achieved, the optimum is also achieved. The likelihood of achieving a feasible solution is significantly higher if the cost incurred by the quadratic expressions of penalties due to relaxed constraints has been accurately weighted by appropriate λ multipliers.



(a) Infeasible solution for the MIS problem.



(b) Optimal (feasible) solution of the MIS problem.

Fig. 5: Example of outcomes from solving the MIS problem: Both a feasible and an infeasible solution are presented. Vertices included in the independent set are highlighted in blue/red. Sub-Figure 5a The red vertex x_3 is adjacent to x_1 and x_4, both of which are included in the independent set. This adjacency violates the independence criterion, rendering the solution infeasible. Sub-Figure 5b The 3 vertices of the independent set do not share any edges.

5 Extension: making a 3-SAT solution with *solQHealer2* by using the 3-$SAT \leq_p MIS$ reduction

A procedure based on *solQHealer*, which we denote as *solQHealer2*, can also be used to search for a solution to the 3-SAT problem. The only difference between the two procedures is that the first seeks feasibility, whereas the second seeks optimality.

In [28], while attempting to solve 3-SAT on a quantum machine, we use the polynomial reduction from 3-SAT to MIS to obtain a less dense graph. This is due to the complexity of the QUBO obtained by its direct formulation from the clauses, compared to the QUBO of MIS, which is obtained more easily. Thus, for quantum machines with static qubit topology, such as D-Wave machines, embedding (i.e., mapping the QUBO graph to the qubit graph) is easier to perform and problems may be solved more efficiently.

The UML diagram presented in Figure 6 summarizes the steps required to find a solution to 3-SAT using *solQHealer2* and the reduction 3-$SAT \leq_p MIS$. An infeasible solution for 3-SAT is first acquired. This solution is then transformed into an MIS solution $\mathcal{S}^{\text{MIS}}$. Next, the process proceeds as previously described, except that the RQA iterations stop when the solution is optimal (i.e., when the number of vertices in the independent set is equal to the number of clauses in the 3-SAT instance), as this is how a solution to 3-SAT is obtained with

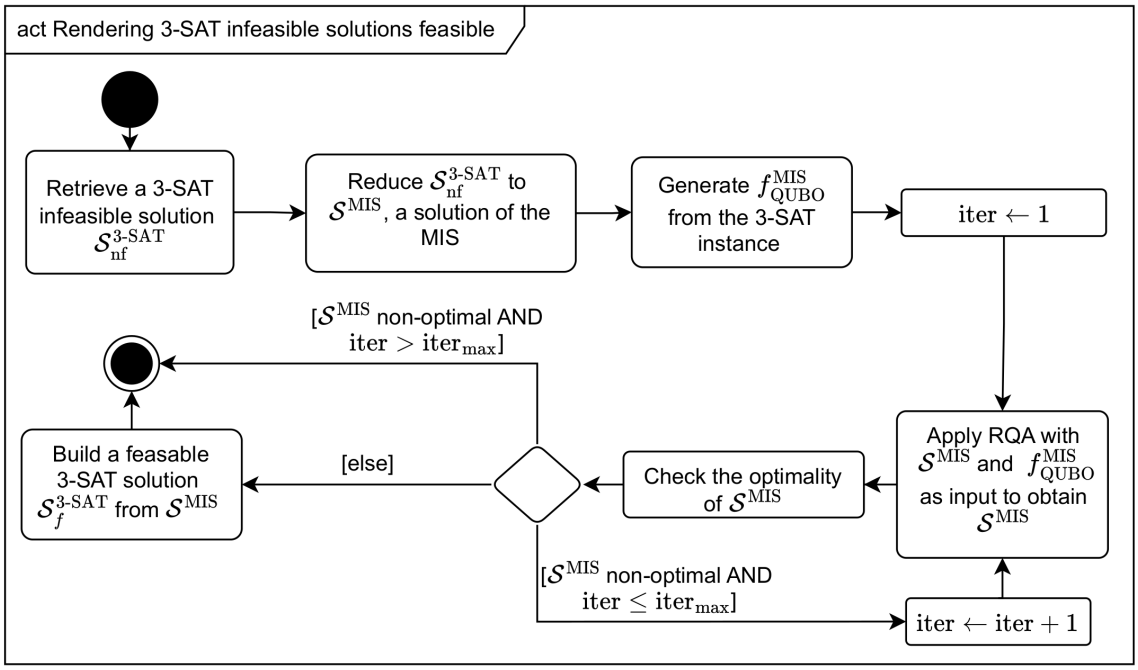


Fig. 6: UML activity diagram: Application of *solQHealer*2 to make a solution feasible for a 3-SAT problem by reducing it in polynomial time to the MIS.

this reduction. Once the optimal MIS solution is obtained, it is then transformed into a solution for the 3-SAT problem. The steps of the reduction 3-$SAT \leq_p MIS$ are provided in the Appendix.

6 Computational Experiments on quantum machine

We generate a set of instances, indexed by $\mathcal{I}$, whose characteristics are given in Table 1, where $f_{\text{opt}}^{\text{MIS}}$ is the objective function of the (feasible) optimal solution. We designate the MIS graph as $\mathcal{G}$ and denote the set of vertices and the set of edges as $\mathcal{V}$ and $\mathcal{E}$, respectively. We consider the following two values for expressing the characteristics of an MIS instance:

- dens$_{\mathcal{G}}$: the density of the graph,
- $\deg_{\mathcal{G}}^{\max}$: the maximum degree of a graph.

$\mathcal{I}$	n^{MIS}	m^{MIS}	$f_{\text{opt}}^{\text{MIS}}$	dens$_{\mathcal{G}}$	$\deg_{\mathcal{G}}^{\max}$
1	30	75	-10	0.17	7
2	45	169	-15	0.17	9
3	60	186	-20	0.11	10
4	75	421	-25	0.15	13
5	90	423	-30	0.11	14

Table 1: Characteristics of the MIS instances.

The degree of a vertex is the number of edges connected to it, and the maximum degree of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is then the highest degree among all vertices in $\mathcal{V}$. We denote by $\deg_i$ the degree of vertex $i : i \in \mathcal{V}$ and we can express the maximum degree of the graph $\deg_{\mathcal{G}}^{\max}$ as follows:

$$\deg_{\mathcal{G}}^{\max} = \max\{\deg_i : i \in \mathcal{V}\}. \tag{7}$$

One of the main issues with D-Wave machines is the embedding, i.e., the process that maps the input graph onto the qubit graph. The connectivity of the latter is limited (e.g., the Advantage machine has over 5000 qubits, however, each qubit is only connected to 15 other qubits). An embedding process is necessary, which implies that the number of qubits used becomes greater than the number of variables. In this paper, embeddings were identified using D-Wave's available heuristic [29]. It is important to note that the problem of embedding is an

active area of research, and that multiple strategies are available in the literature (see, for instance, Okada et al. [30] and Bernal et al. [31]). For the problem considered here, the solved instances can be embedded within the Pegasus topology, as the graphs associated with their QUBO formulation exhibit relatively low density. For instance, a 90-variable instance addressed in this work requires 316 qubits (see Figure 7).

The density of an undirected graph is a measure that is obtained by dividing the number of edges of $\mathcal{G}$ and the number of edges of the related complete graph based on the vertices of $\mathcal{V}$. The density is bounded by $\text{dens}_{\mathcal{G}} = 0$ if $\mathcal{E}$ is empty and by $\text{dens}_{\mathcal{G}} = 1$ if $\mathcal{G}$ is complete. If n^{MIS} denotes the number of vertices and m^{MIS} denotes the number of edges, the density $\text{dens}_{\mathcal{G}}$ is calculated as:

$$\text{dens}_{\mathcal{G}} = 2m^{\text{MIS}}/(n^{\text{MIS}} * (n^{\text{MIS}} - 1)). \tag{8}$$

For all experiments, we used the parameter values given in Table 2 to configure the D-Wave machine.

Parameter	**Value**
Solver	Advantage 6.4
Anneal time T_a	10 μs
Number of anneals N_a	100 ($n^{\text{MIS}} \leq 60$) / 1000 ($60 < n^{\text{MIS}} \leq 75$) / 2500 ($n^{\text{MIS}} = 90$)
Chain strength value C_s	1000
Maximum number of iterations $\text{iter}_{\max}$	100

Table 2: Parameters of the D-Wave machine.

In these experiments, we evaluate the three methods by reporting the number of qubits required, the execution time, and the number of RQA replications needed to reach a feasible solution. We also compute the distance between the obtained solutions and the initial infeasible one using cosine similarity. It measures the angular similarity between two binary vectors by computing the cosine of the angle between them in a high-dimensional space. Since our solution vectors are binary, the cosine similarity $\mathcal{C}$ is defined as:

$$\mathcal{C} = \cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$$

where $\mathbf{u}$ and $\mathbf{v}$ are binary vectors. The resulting value lies in the interval $[0, 1]$, with 1 indicating identical vectors, and 0 indicating orthogonality (i.e., no vertices are shared between the two independent sets represented by the solutions).

In addition, we introduce the following notation:

- $\mathcal{M}$: index of the method used (1, 2, or 3).
- ζ: the level of infeasibility of the first solution considered, which corresponds to the number of edges for which both vertices have been selected in the independent set.
- $\text{iter}^{\text{RQA}}_{\text{best}}$: the number of iterations the procedure needs in order to make the solution feasible, i.e., $\zeta \leq 0$.
- $\text{iter}^{\text{RQA}}_{\text{series}}$: the series of iteration indices for which feasibility was improved; the series of iterations that generate a solution with decreased ζ.
- $f^{\text{MIS}}_{\text{series}}$: the series of objective function values for each iteration in $\text{iter}^{\text{RQA}}_{\text{series}}$.
- $f^{\text{MIS}}_{\text{SA}}$: the objective function value obtained by the Simulated Annealing metaheuristic.
- $\mathcal{Q}$: the number of qubits used to obtain the best solution.
- $\mathcal{T}$: the total QPU time in seconds for all the iterations of RQA until a feasible solution is found or $\text{iter}^{\text{RQA}}_{\max}$ is reached.
- $\mathcal{C}_{\text{series}}$: the series of cosine similarities, computed across RQA replications, between the objective function value of $S^{\mathcal{P}}_{\text{inf}}$ and the best solution currently found by the method $\mathcal{M}$ being executed.

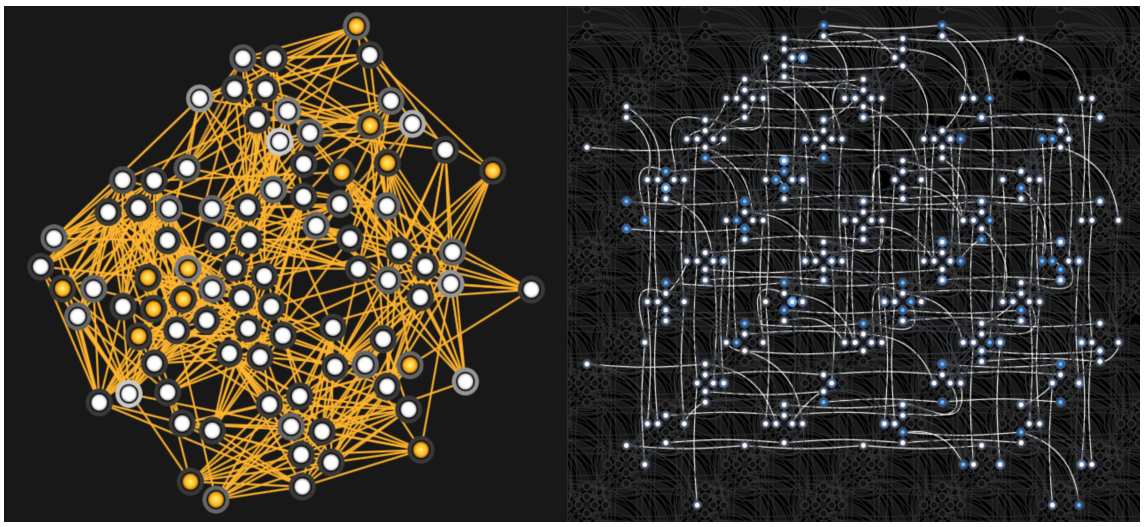


Fig. 7: Example of an embedding of a MIS instance obtained using the Minorminer heuristic [29]. The instance, shown on the left, contains 90 vertices. Its embedding into the Pegasus topology of the Advantage 6.4 quantum processing unit is depicted on the right, requiring 316 qubits.

Table 3 presents the results obtained in solving the 5 instances of $\mathcal{I}$. The goal here is not to weigh the multipliers in a way that facilitates exploration towards the optimal solution, but to seek feasibility. Therefore, we do not analyse the gap with the optimal value, as `solQHealer` stops as soon as a feasible solution is found. In parallel, we run the Simulated Annealing algorithm (as implemented by D-Wave) on the same QUBO to provide a comparative baseline for contextualizing the experiments. No feasible solution was found by Simulated Annealing (all $f_{\text{SA}}^{\text{MIS}}$ values are positive) for any instance when using the same number of anneals as in each RQA execution.

Each of the three methods starts from an infeasible solution whose objective function value is equal to $f_{\text{SA}}^{\text{MIS}}$. The level of infeasibility ζ ranges from one to seven, meaning that, for the MIS, the vertices of one to seven edges were selected. It is noteworthy that the penalty weights of λ allow us to deduce the feasibility of the new solutions once a negative objective value has been reached. In all these experiments, all five infeasible solutions became feasible with the Method 1 and the Method 3.

$\mathcal{I}$	ζ	λ	$\mathcal{M}$	$f_{\text{SA}}^{\text{MIS}}$	$f_{\text{series}}^{\text{MIS}}$ ($f_{\text{inf}}^{\text{MIS}}$ is the first)	$\text{iter}_{\text{series}}^{\text{RQA}}$	$\mathcal{Q}$	$\mathcal{T}$
1	1	30	1	42	82→54→21→ **-7**	0→1→2→7	44	0.11
			2		82→ **-4**	0→1		0.02
			3		82→53→22→21→ **-6**	0→2→3→5→12		0.21
2	4	45	1	24	257→212→34→ **-5**	0→1→2→40	103	0.69
			2		257→213→122→32→ inf	0→1→2→5→100		1.73
			3		257→170→78→34→ **-9**	0→1→2→4→91		1.57
3	2	60	1	22	341→105→48→ **-15**	0→1→18→24	114	0.42
			2		341→227→224→45→ inf	0→1→2→5→100		1.72
			3		341→293→225→167→102→ **-12**	0→1→3→4→10→32		0.57
4	4	75	1	330	208→131→130→57→54→ **-20**	0→1→9→12→23→36	171	6.80
			2		208→205→134→129→ **-17**	0→4→5→11→12		3.03
			3		208→133→131→61→58→ **-16**	0→1→2→22→66→67		13.96
5	7	90	1	655	155→68→65→ **-22**	0→6→42→43	316	19.44
			2		155→67→66→65→ inf	0→3→5→24→100		44.89
			3		155→72→67→65→ **-26**	0→18→20→29→48		24.88

Table 3: Results obtained for the three proposed methods on the five considered instances. The objective function values $f_{\text{series}}^{\text{MIS}}$ of the feasible solutions found are highlighted in bold.

Table 4 reports the cosine similarities obtained during the execution of the three methods across the five instances. Methods 1 and 3 appear to perform best in this regard, although Method 3 produces a feasible solution with zero cosine similarity for the first instance (i.e., no common vertices between the independent sets). Conversely, for the most difficult instance, the same method yields a solution with a cosine similarity of 0.36.

$\mathcal{I}$	$\mathcal{M}$	$\mathcal{C}_{\text{series}}$
1	1	1→0.43→0.35→ **0.27**
	2	1→ **0.18**
	3	0.27→0.37→0.24→ **0.00**
2	1	1→0.46→0.17→ **0.12**
	2	1→0.56→0.38→0.23→ inf
	3	1→0.09→0.24→0.08→ **0.18**
3	1	1→0.18→0.26→ **0.41**
	2	1→0.13→0.40→0.36→ inf
	3	1→0.39→0.24→0.32→0.49→ **0.46**
4	1	1→0.28→0.16→0.11→0.11→ **0.22**
	2	1→0.16→0.06→0.26→ **0.24**
	3	1→0.12→0.22→0.26→0.35→ **0.18**
5	1	1→0.38→0.20→ **0.17**
	2	1→0.29→0.37→0.32→ inf
	3	1→0.19→0.21→0.24→ **0.36**

Table 4: Cosine similarities obtained for the three proposed methods on the five considered instances. Values corresponding to feasible solutions are highlighted in bold.

7 Discussion

In this work, we explored the practical implementation of a new procedure that uses Reverse Quantum Annealing (RQA) to transform infeasible solutions into feasible ones within a limited time frame. Such a procedure is useful for any setting in which fast, online optimization is required (e.g., train scheduling, bin packing, etc.). To the best of our knowledge, we are the first to address the repair of infeasible solutions via RQA, marking a significant contribution to the integration of quantum computing in the field of operations research. We provide a QUBO model for instances of the 3-SAT problem formulated as an MIS problem, but the procedure described in this work can be used for any other optimization problem.

Today, several types of quantum machines are capable of accepting a warm-start solution as input, in addition to the optimization model. This is particularly true for analog adiabatic quantum machines (e.g., D-Wave or Pasqal), which, through QUBO formulations, can initiate their execution not from an uniform superposition state of the qubits (i.e., where the probability of measuring 0 or 1 is equal), but rather from a state corresponding to a given initial solution. The solQHealer approach can be adapted to such systems, and the work presented in this paper (especially the computation of cosine similarities) may help identify the quantum hardware that remains closest to the initially provided solution.

Similar to QA, the quantum annealing process of RQA can be finetuned by adjusting various annealing parameters [18]. It would thus be interesting to study the impact of these parameters on the performance of finding a feasible a solution. Moreover, RQA is currently only possible on D-Wave machines, but other analog quantum machines are also capable of starting from an existing solution. Notably this is the case for Pasqal's neutral atoms quantum machines [32], thus it would be interesting to study the performance of our method on a different quantum architecture.

Declarations

In the following sections we provide further information concerning, data and code availability as well as other relevant matters.

Data Availability

The 3-SAT original instances used in this work, the code that transforms them into MIS, and the results obtained from their resolution using SA, and RQA can be downloaded from the following public repository below.

https://github.com/ceche1212/solQHealer

Code Availability

All the code, used in the elaboration of this work, can be downloaded from the the following public reposory below.

https://github.com/ceche1212/solQHealer

In order to run the RQA methods, the user must provide a valid D-Wave API token, in the different functions that used them, inside the `MIS_Healer_Token.py` file.

Acknowledgements

This research benefited from the support of the FMJH Program PGMO.

Author contribution

S.D: conceptualization, investigation, writing—original draft; L.F.P.A & S.C : conceptualization, writing—review and editing. The three authors gave their final approval for publication and agreed to be held accountable for the work performed therein.

Ethics & Conflict of interest declarations

The authors have no Conflict of interest to declare that are relevant to the content of this article.

Additional information

Correspondence and requests for materials should be addressed to S.D.

Appendix A The 3-$SAT \leq_p$ MIS reduction

Let $\mathcal{C}$ denotes a set of clauses and let v_j denote a binary variable for $j = 1, \ldots, |\mathcal{V}|$, where $\neg v_j$ is used to denote its negation. In addition, define binary variable $y_{i,j}$ for $i = 1, \ldots, |\mathcal{C}|$ and $j = 1, \ldots, |\mathcal{V}|$, independent of whether the variable is negated:

$$y_{i,j} = \begin{cases} 1 \text{ if the vertex } (i,j) \text{ is in the independent set } \mathcal{S}, \\ 0 \text{ otherwise.} \end{cases} \tag{A1}$$

Each variable $y_{i,j}$ is represented by a vertex in the set $\mathcal{V}$ of a graph $\mathcal{G}$. An edge e in the set $\mathcal{E}$ exists for one of the following reasons:

- The two connected variables belong to the same clause i, for $i = 1, \ldots, |\mathcal{C}|$.
- The two connected variables correspond to the same original variable j, for $j = 1, \ldots, |\mathcal{V}|$, with one being the negation of the other.

If the MIS resolution on the graph $\mathcal{G}$ yields an optimal solution, a corresponding solution to the 3-SAT problem is considered to exist when the size of set $\mathcal{S}$ matches the total number of clauses. If this condition is not met, it implies that the given 3-SAT instance is unsatisfiable. Specifically, an optimal MIS solution must feature exactly one $y_{i,j}$ that is set to 1 within each 3-vertex clique i to ensure that all clauses are satisfied. Given the presence of such an optimal solution, the values of the 3-SAT variables v_j, for $j = 1$ to $|\mathcal{V}|$, can be inferred from each instance of $y_{i,j} = 1$ as follows:

$$v_j = \begin{cases} 0 \text{ if the variable } v_j \text{ appears as a negation in clause } i, \\ 1 \text{ otherwise.} \end{cases} \tag{A2}$$

Appendix B Feasibility tradeoff in SA and RQA

When compared with SA, RQA has various characteristics that might be considered advantageous in the context of transforming infeasible solutions into feasible ones.

Simulated annealing is renowned for its flexibility and capacity to escape local optima by accepting worse solutions under the influence of a temperature parameter. However, when starting from a warm-start solution that is infeasible, regulating the algorithm's behavior purely through temperature introduces a set of intricate tradeoffs, especially between exploration and exploitation, and between feasibility restoration and objective optimization.

The acceptance of new solutions in simulated annealing is guided by the Metropolis criterion, given by $\exp\left(-\frac{\Delta E}{T}\right)$, where ΔE denotes the difference in fitness between the incumbent solution and the newly proposed one, and T represents the current temperature. This criterion, originally inspired by the behavior of physical systems undergoing thermodynamic transitions, forms the basis of the Metropolis–Hastings algorithm [33]. According to this rule, a new solution is accepted if $W > q$, where q is a random number drawn from a continuous uniform distribution in the interval $[0, 1]$. At high temperatures, the acceptance probability increases, making the algorithm more permissive and more likely to accept worse or even infeasible solutions. This behavior is illustrated in Figure B1 below.

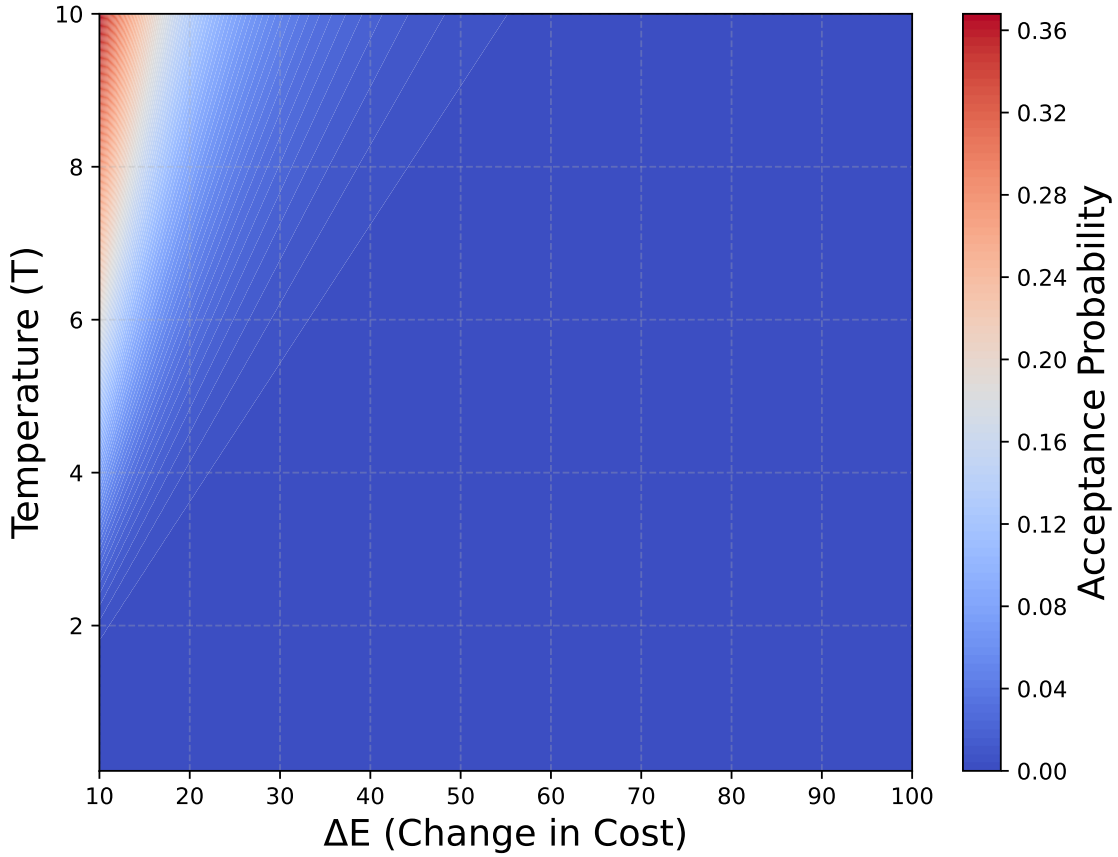


Fig. B1: Probability of accepting an infeasible solution, based energy and temperature

According to this acceptance rule and under the typical implementation of simulated annealing, an infeasible solution has only a slim, almost negligible chance of being accepted except when the temperature T is high. At high temperatures, the Metropolis criterion becomes more permissive: the probability of accepting worse solutions, including infeasible ones, approaches to one. This encourages broad exploration of the search space and allows the algorithm to potentially "repair" infeasibilities by moving toward feasible regions. However, this same property introduces a destabilizing randomness. The algorithm may behave like a random sampler, readily accepting not only infeasible solutions but also sequences of low-quality solutions with little regard for search direction. In this regime, any meaningful structure or partial quality embedded in the initial warm-start solution can quickly erode. Without anchoring to the informative aspects of the starting point, the algorithm may drift arbitrarily through the search space, ultimately losing the advantages offered by the warm-start.

In the typical implementation of simulated annealing, the search for feasible solutions is primarily guided by incorporating feasibility penalties into the objective function. These penalties assign high costs to constraint violations, thereby redirecting the algorithm's focus toward satisfying constraints and implicitly reshaping the optimization landscape. However, this adjustment introduces a new challenge: as penalties increase, the Metropolis acceptance probability becomes highly sensitive to constraint violations. In extreme cases, large penalties can render any infeasible move overwhelmingly unfavorable, especially as the temperature decreases. This suppresses the algorithm's ability to explore and repair an infeasible initial solution, and increases the risk of getting trapped in regions far from the provided warm start.

Conversely, if the infeasibility penalties are too small, the cost landscape fails to meaningfully distinguish between repairing infeasibilities and improving the objective function. In such cases, the SA algorithm may lack clear direction toward feasibility and instead spend excessive time optimizing within infeasible regions. This creates ambiguity in the algorithm's decision-making, making it difficult to determine whether to prioritize constraint satisfaction or cost minimization. The resulting behavior can become oscillatory or unfocused. The problem becomes even more pronounced when multiple constraints are violated in the warm-start solution, as illustrated in Figure B2 below.

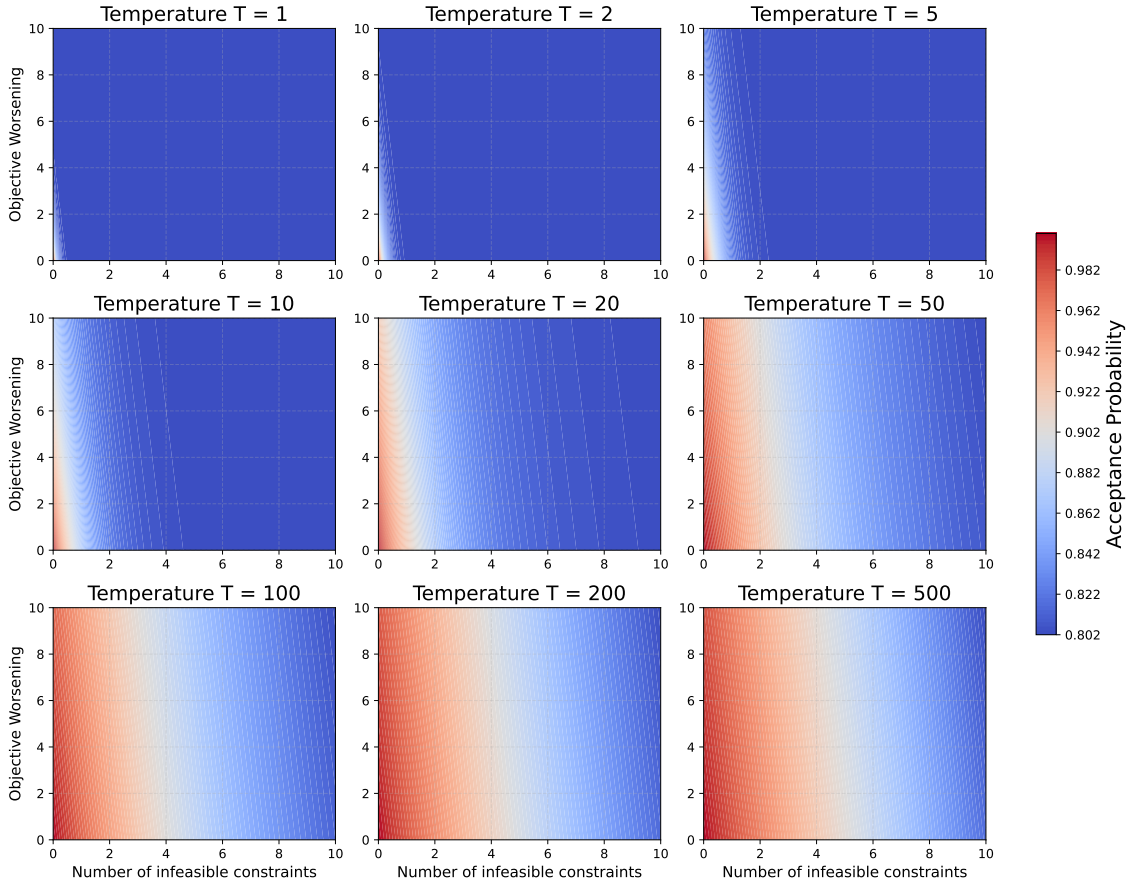
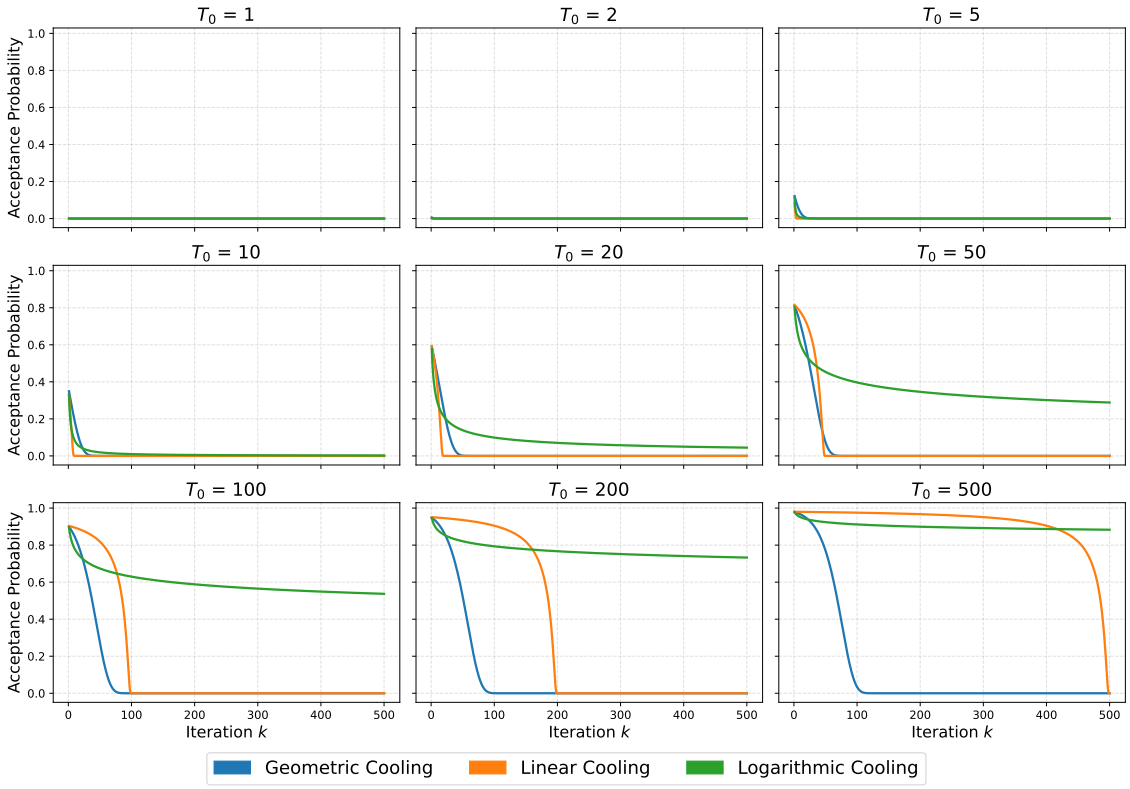


Fig. B2: Objective and feasibility penalty tradeoff for different temperatures in SA

Repairing multiple dimensions of infeasibility typically requires extended periods of *high-temperature sampling*, during which the algorithm maintains the flexibility to cross infeasible valleys. However, standard cooling schedules; linear, geometric, or exponential; do not naturally accommodate prolonged high-temperature phases unless explicitly slowed down (i.e., very low decay rates). Yet, doing so compromises the *exploitation phase* that SA relies on in its later stages. A slow decay maintains randomness too long, while a fast decay inhibits constraint repair before the system "freezes".

This brings us to a deeper insight: temperature alone is a powerful mechanism yet, insufficient to regulate the conflicting dynamics of feasibility recovery, feature preservation, and optimization. If one aims to retain valuable characteristics of a warm-start solution; such as solution structure, good sub components, or domain-specific logic, while transitioning toward feasibility, temperature must be complemented with mechanisms that guide the search more deliberately. These could include constraint-specific repair heuristics, adaptive penalty



H

Fig. B3: Evolution of acceptance probability for infeasible solution, for different cooling mechanism of SA and different temperatures

schemes, or hybrid strategies that alternate between targeted feasibility phases and objective-driven refinement. This is illustrated in Figure B3 below.

In short, relying solely on temperature to balance exploration, exploitation, and feasibility introduces a "*tug-of-war*" of competing forces. High temperatures may allow for feasibility repair but threaten meaningful structure retention. Low temperatures foster exploitation but make constraint violations nearly untouchable. Without additional guidance, temperature becomes a blunt instrument; too coarse to finely control the rich, multi-faceted dynamics involved in transforming an infeasible warm start into a high-quality feasible solution.

References

[1] Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse ising model. Physical Review E **58**(5), 5355 (1998)

[2] Vert, D., Sirdey, R., Louise, S.: Benchmarking quantum annealing against "hard" instances of the bipartite matching problem. SN Computer Science **2**, 1–12 (2021)

[3] Bertsimas, D., Jaillet, P., Martin, S.: Online vehicle routing: The edge of optimization in large-scale applications. Operations Research **67**(1), 143–162 (2019)

[4] Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A., Preda, D.: A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. Science **292**(5516), 472–475 (2001)

[5] Tasseff, B., Albash, T., Morrell, Z., Vuffray, M., Lokhov, A.Y., Misra, S., Coffrin, C.: On the emerging potential of quantum annealing hardware for combinatorial optimization. Journal of Heuristics **30**(5), 325–358 (2024)

[6] Creemers, S., Armas, L.F.P.: Discrete optimization: A quantum revolution? European Journal of Operational Research **323**(2), 378–408 (2025)

[7] Kirkpatrick, S., Gelatt Jr, C.D., Vecchi, M.P.: Optimization by simulated annealing. science **220**(4598), 671–680 (1983)

[8] Albash, T., Lidar, D.A.: Adiabatic quantum computation. Reviews of Modern Physics **90**(1), 015002 (2018)

[9] Born, M., Fock, V.: Beweis des adiabatensatzes. Zeitschrift für Physik **51**(3-4), 165–180 (1928)

[10] Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014)

[11] King, A.D., Nocera, A., Rams, M.M., Dziarmaga, J., Wiersema, R., Bernoudy, W., Raymond, J., Kaushal, N., Heinsdorf, N., Harris, R., Boothby, K., Altomare, F., Asad, M., Berkley, A.J., Boschnak, M., Chern, K., Christiani, H., Cibere, S., Connor, J., Dehn, M.H., Deshpande, R., Ejtemaee, S., Farre, P., Hamer, K., Hoskinson, E., Huang, S., Johnson, M.W., Kortas, S., Ladizinsky, E., Lanting, T., Lai, T., Li, R., MacDonald, A.J.R., Marsden, G., McGeoch, C.C., Molavi, R., Oh, T., Neufeld, R., Norouzpour, M., Pasvolsky, J., Poitras, P., Poulin-Lamarre, G., Prescott, T., Reis, M., Rich, C., Samani, M., Sheldan, B., Smirnov, A., Sterpka, E., Clavera, B.T., Tsai, N., Volkmann, M., Whiticar, A.M., Whittaker, J.D., Wilkinson, W., Yao, J., Yi, T.J., Sandvik, A.W., Alvarez, G., Melko, R.G., Carrasquilla, J., Franz, M., Amin, M.H.: Beyond-classical computation in quantum simulation. Science **0**(0), 6285 (2025) https://doi.org/10.1126/science.ado6285 https://www.science.org/doi/pdf/10.1126/science.ado6285

[12] Baldassi, C., Zecchina, R.: Efficiency of quantum vs. classical annealing in nonconvex learning problems. Proceedings of the National Academy of Sciences **115**(7), 1457–1462 (2018)

[13] Istrail, S.: Statistical mechanics, three-dimensionality and np-completeness: I. universality of intracatability for the partition function of the ising model across non-planar surfaces. In: Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing, pp. 87–96 (2000)

[14] Venturelli, D., Kondratyev, A.: Reverse quantum annealing approach to portfolio optimization problems. Quantum Machine Intelligence **1**(1-2), 17–30 (2019)

[15] Ikeda, K., Nakamura, Y., Humble, T.S.: Application of quantum annealing to nurse scheduling problem. Scientific reports **9**(1), 12837 (2019)

[16] Rocutto, L., Destri, C., Prati, E.: Quantum semantic learning by reverse annealing of an adiabatic quantum computer. Advanced Quantum Technologies **4**(2), 2000133 (2021)

[17] Golden, J., O'Malley, D.: Reverse annealing for nonnegative/binary matrix factorization. Plos one **16**(1), 0244026 (2021)

[18] Venturelli, D., Marchand, D.J., Rojo, G.: Quantum annealing implementation of job-shop scheduling. arXiv

preprint arXiv:1506.08479 (2015)

[19] Karp, R.M.: Reducibility among combinatorial problems, complexity of computer computations (proc. sympos., ibm thomas j. watson res. center, yorktown heights, ny, 1972). MR **378476**(51), 14644 (1972)

[20] Karp, R.M., Wigderson, A.: A fast parallel algorithm for the maximal independent set problem. Journal of the ACM (JACM) **32**(4), 762–773 (1985)

[21] Garey, M.R., Johnson, D.S.: "strong"np-completeness results: Motivation, examples, and implications. Journal of the ACM (JACM) **25**(3), 499–508 (1978)

[22] Goddard, W., Henning, M.A.: Independent domination in graphs: A survey and recent results. Discrete Mathematics **313**(7), 839–854 (2013)

[23] Lucas, A.: Ising formulations of many np problems. Frontiers in physics **2**, 5 (2014)

[24] Glover, F., Kochenberger, G., Hennig, R., Du, Y.: Quantum bridge analytics i: a tutorial on formulating and using qubo models. Annals of Operations Research **314**(1), 141–183 (2022)

[25] Glover, F., Kochenberger, G., Ma, M., Du, Y.: Quantum bridge analytics ii: Qubo-plus, network optimization and combinatorial chaining for asset exchange. Annals of Operations Research **314**(1), 185–212 (2022)

[26] Sadeghi, H.: Quantum computing with neutral atoms with applications in energy. In: EAGE Seventh High Performance Computing Workshop, vol. 2023, pp. 1–3 (2023). European Association of Geoscientists & Engineers

[27] Silva Coelho, W., Henriet, L., Henry, L.-P.: Quantum pricing-based column-generation framework for hard combinatorial problems. Physical Review A **107**(3), 032426 (2023)

[28] Deleplanque, S.: Solving 3SAT and MIS problems with analog quantum machines. In: Computational Science and Its Applications – ICCSA 2023 Workshops, pp. 429–439. LNCS, Springer Nature Switzerland, Cham (2023)

[29] Cai, J., Macready, W.G., Roy, A.: A practical heuristic for finding graph minors. arXiv preprint arXiv:1406.2741 (2014)

[30] Okada, S., Ohzeki, M., Terabe, M., Taguchi, S.: Improving solutions by embedding larger subproblems in a d-wave quantum annealer. Scientific reports **9**(1), 2098 (2019)

[31] Bernal, D.E., Booth, K.E., Dridi, R., Alghassi, H., Tayur, S., Venturelli, D.: Integer programming techniques for minor-embedding in quantum annealers. In: Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 17th International Conference, CPAIOR 2020, Vienna, Austria, September 21–24, 2020, Proceedings 17, pp. 112–129 (2020). Springer

[32] Henriet, L., Beguin, L., Signoles, A., Lahaye, T., Browaeys, A., Reymond, G.-O., Jurczak, C.: Quantum computing with neutral atoms. Quantum **4**, 327 (2020)

[33] Hastings, W.K.: Monte carlo sampling methods using markov chains and their applications. Biometrika **57**(1), 97–109 (1970). Accessed 2024-07-11