

Louvain School of Management

The size dependence of QUBO penalty parameters: calibrating the Quadratic Knapsack Problem on physical hardware

Author: **Nicolas NOPENAIRE**
Supervisor: **Prof. Stefan CREEMERS**
Academic year 2025–2026
Master [120] in Business Engineering, professional focus
Major Business Analytics

STATEMENTS ON YOUR USE OF GENERATIVE AI

Confirm the following statements:

Statements on responsible use of generative AI	Your answer
The content of my/our master's thesis is my/our original output, even if I/we used generative AI to help prepare it.	☒ Yes ☐ No
I/we master the theories, tools and methods that I/we use for the thesis.	☒ Yes ☐ No
I/we verified that the output presented in the thesis is accurate and valid, and that it contains no hallucination, no false reference, etc.	☒ Yes ☐ No
I/we acknowledge that I/we master the final output and that I am/we are able to explain it and answer questions about it.	☒ Yes ☐ No
I/we made sure not to provide generative AI tools with access to confidential data or information	☒ Yes ☐ No

SIGNATURE

Sign your declaration:

Author last name, first name(s)	Date	Signature
1 Nopenaire, Nicolas	05 aug. 2026	

X

Nicolas NOPENAIRE
Student

During the preparation of this master's thesis, I, Nicolas Nopenaire, used Claude (Anthropic), Consensus, Gemini (Google), Perplexity, Grammarly, DeepL and Reverso for the following purposes:

- **Writing assistance:** structuring the manuscript, reformulating passages, checking consistency between chapters, and correcting linguistic errors (Claude, Grammarly, DeepL, Reverso).
- **Literature search and source discovery:** identifying candidate references and locating supporting literature, subsequently verified against the original publications (Consensus, Perplexity).
- **Python and L^AT_EX code assistance:** debugging Python scripts, resolving L^AT_EX compilation errors, and typographic formatting (Claude, Gemini).
- **Proofreading:** critical rereading of the text and detection of inconsistencies in formulation (Gemini, Claude).

Every reference cited in the thesis was verified against its original publication, and no numerical value reported was produced by these tools. By signing this declaration, I confirm that the content of this master's thesis reflects my original work, augmented by the responsible use of AI.

Abstract

Solving a constrained combinatorial problem on a quantum annealer requires absorbing the constraint into the objective as a penalty term, and the coefficient of that term decides whether the hardware returns anything usable. For the Quadratic Knapsack Problem the rules for choosing it have been characterized almost entirely in simulation, where minor embedding, chain integrity and finite coupler precision do not exist. This thesis measures what those rules are worth on a physical D-Wave processor.

Three protocols divide the question. A log-linear regression fitted on 1 437 calibrations against a classical exact oracle returns a size exponent of 0.972 (s.e. ± 0.034 , clustered by design cell) where the leading published rule reports 0.09, a disagreement of an order of magnitude established with no hardware involved and consistent with QPU campaigns at three instance sizes. That rule is fitted on a coefficient optimal under classical repair post-processing, where the quantity a processor needs is the penalty that enforces the constraint unaided. The slack encoding of the capacity constraint returns no viable run across 311 attempts on two topologies, failing on Zephyr because its auxiliary variables exceed the largest embeddable clique and on Pegasus, where the embedding succeeds, because no penalty value brings the solution weight down to capacity before the chains break. Transfer between instances works by region rather than by point, a single calibrated pair overrunning capacity on two instances of six and filling the knapsack to eight per cent of its capacity on a third, while the fitted rule stops working altogether above 25% density.

Every one of the ten Billionnet & Soutif instances at $n = 100$ and density 25% nonetheless admits a penalty setting under which the processor returns a feasible solution with reliable chains, and eight reach the proven optimum exactly. One methodological result cuts across the three protocols: the same grid supports opposite conclusions about the quadratic penalty coefficient depending on whether a chain-break reliability criterion is applied, which bears on how benchmark results on annealing hardware should be reported.

Keywords: quadratic knapsack problem, quantum annealing, QUBO, penalty calibration, minor embedding, D-Wave

Acknowledgments

I thank Professor Stefan Creemers, who supervised this work and whose availability throughout never wavered, and Professor Samuel Deleplanque, whose guidance on quantum programming and on D-Wave hardware made the experimental part of this thesis possible.

To Eva Martin, for her support through every stage of this work.

To my family and to my parents-in-law, for their encouragement, and to my friends, for the balance they provided along the way.

Contents

1	Introduction	1
2	Literature Review	2
2.1	The Quadratic Knapsack Problem	2
2.1.1	From the linear to the quadratic knapsack problem	2
2.1.2	Compact formulation and instance parameters	3
2.1.3	Reference instances	4
2.2	Classical Resolution of the QKP	4
2.3	Quantum resolution of the QKP	5
2.3.1	Gate-based search and its current limits	5
2.3.2	The physics of Quantum Annealing	6
2.3.3	Structural alignment between the QKP and Quantum Annealing	6
2.3.4	The hardware processor topologies: Advantage and Advantage2	7
2.3.5	Embedding, chain integrity, and coupler precision	7
2.4	QUBO formulation of the QKP and the calibration problem	8
2.4.1	From a constrained problem to an unconstrained one	8
2.4.2	The difficulty of an inequality constraint	9
2.4.3	The slack formulation	9
2.4.4	The slack-free formulation: unbalanced penalization	9
2.4.5	Choosing the slack-free formulation	10
2.4.6	The calibration problem	10
2.5	State of the art in parameter transfer	11
2.5.1	The near-invariant scaling law of Ohno et al. (2024)	11
2.5.2	The linear-in-size heuristic of Parizy and Togawa (2021)	12
2.5.3	The slack-free penalty of Montanez-Barrera et al. (2024)	12
2.5.4	What simulation and hybrid solvers leave open	12
2.6	What the survey leaves unsettled	13
3	Methodology	13
3.1	Instances and ground truth	14
3.1.1	The Billionnet published benchmark	14
3.1.2	Generated calibration sets	14
3.1.3	Ground truth by regime	15
3.2	QUBO construction and normalization	15
3.2.1	From the instance to the BQM	15
3.2.2	Physical and dimensionless penalties	15
3.3	Hardware protocol	16
3.3.1	Sampling	16
3.4	Calibration procedures	17
3.4.1	Nelder-Mead calibration with fixed κ_2	17
3.4.2	Regression and the failed transfer to Billionnet-scale instances	18
3.4.3	Empirical anchoring: Method A	19
3.4.4	Two-dimensional grid search	20
3.5	Evaluation metrics	21
3.6	Falsification protocols	21
3.6.1	P1 — The scaling law of Ohno et al.	21
3.6.2	P2 — The slack wall on Zephyr	22

	3.6.3	P3 — Parameter mapping and the transfer of calibrated parameters	22
	3.7	Scope	23
4	Results		23
	4.1	Overview and reading guide	23
	4.2	P1 — The scaling exponent on instance size	23
	4.3	P2 — The slack wall: two levels of obstruction	25
	4.3.1	The four configurations	26
	4.3.2	Refining the topological prediction	27
	4.3.3	The electronic wall on Pegasus	27
	4.4	P3 — Parameter transfer on physical hardware	28
	4.4.1	The chain-break gate as a discriminator	29
	4.4.2	Transfer across instances	30
	4.4.3	Transfer from classical calibration to hardware	31
	4.4.4	Transfer to other densities	32
	4.4.5	The tightness correction in retrospect	33
	4.5	Hardware characterization	34
	4.6	Chapter summary	34
5	Discussion		35
	5.1	The nature of the scaling exponent	35
	5.2	The instance space is not uniform	37
	5.3	The chain-break gate as an evaluative metric	38
	5.4	The calibration architecture as a general strategy	38
	5.5	Limitations	39
	5.6	Future work	39
6	Conclusion		40
7	Bibliography		43
A	Appendix		48
	A.1	Connectivity and embedding of the QUBO-encoded QKP	48
	A.1.1	Density-independent connectivity of the penalty term	48
	A.1.2	The chain Hamiltonian of minor embedding	48
	A.2	Complete penalty sweep, slack formulation on Pegasus	50
	A.3	Instance generators	52
	A.4	Pipeline verification	52
	A.5	Grid coverage per instance	53
	A.6	Method A, complete results	53
	A.7	Regression diagnostics	54
	A.8	Data and code availability	54

List of Abbreviations

Abbreviation	Meaning	Page
BQM	Binary Quadratic Model	13
CBF	Chain-Break Fraction	8
CPU	Central Processing Unit	5
CQM	Constrained Quadratic Model	13
DP	Dynamic Programming	2
GPU	Graphics Processing Unit	1
KP01	Binary 0–1 Knapsack Problem	2
MILP	Mixed-Integer Linear Programming	4
NISQ	Noisy Intermediate-Scale Quantum	6
NP	Nondeterministic Polynomial time	1
QAOA	Quantum Approximate Optimization Algorithm	12
QKP	Quadratic Knapsack Problem	1
QPU	Quantum Processing Unit	1
QUBO	Quadratic Unconstrained Binary Optimization	2
UTC	Uniform Torque Compensation	16

List of Symbols

Symbol	Description
n	Number of items / problem size (variables)
d	Instance density ($d \in [0, 1]$)
α	Tightness ratio ($\alpha = C / \sum_i w_i$)
C	Knapsack capacity constraint
w_i	Weight of item i
p_i	Linear profit of item i
p_{ij}	Pairwise (quadratic) profit between items i and j
λ	Penalty coefficient (raw / physical)
κ_1	Dimensionless linear penalty coefficient
κ_2	Dimensionless quadratic penalty coefficient
c_n	Size scaling exponent
c_d	Density scaling exponent
c_α	Tightness scaling exponent
J_F	Ferromagnetic chain coupling strength
Pct_Opt	Ratio of best objective value to proven optimum (%)
$\sigma_{\log}$	Logarithmic standard deviation of regression residuals
$P_{\max}$	Largest objective coefficient, $\max_{ij} v_{ij} $
$W_{\max}$	Largest item weight, $\max_i w_i$
$h(x)$	Constraint slack, $C - \sum_i w_i x_i$
$\tilde{h}(x)$	Constraint slack in units of $W_{\max}$
A	Prefactor of the scaling law
τ	Heuristic rescaling of Ohno et al., $(d/100)\sqrt{1/\alpha}$
$ S $	Number of items in the proven optimum
N_c	Chain length, in physical qubits

1 Introduction

In the Quadratic Knapsack Problem (QKP) a container of fixed capacity receives a subset of items, each carrying a weight and an individual profit, with a further profit collected whenever two selected items are paired. Formally defined by Gallo et al. (1980) after a decade of applications in facility location and portfolio selection, the problem is strongly NP-hard and has resisted exact solution beyond a few hundred variables at low density (Galli et al., 2025). Its quadratic objective also sets it apart among classical benchmarks, every exact method reviewed in Chapter 2 either linearizing it at a cost of $O(n^2)$ auxiliary variables or enumerating the 2^n subsets its dynamic program degenerates into.

Quantum annealing offers a route that pays neither price. A D-Wave processor minimizes a quadratic unconstrained binary objective, which the QKP already is, so the form in which the problem is stated and the form the hardware minimizes coincide. What does not carry over is the capacity constraint. An annealer recognizes no constraint, and the inequality has to be absorbed into the objective as a penalty term whose coefficient decides everything. Set too low, the objective overruns the capacity and the sampler returns solutions that violate it. Set too high, the penalty flattens the energy landscape until good feasible states become indistinguishable from bad ones, and on a physical processor it competes for coupler precision with the ferromagnetic chains that carry the logical variables.

Choosing that coefficient is the calibration problem, and the literature has approached it almost entirely in simulation. Ohno et al. (2024) fit the first analytical rule for the QKP, a power law in instance size, density and tightness, on simulated annealing and a GPU-based Ising machine. A linear-in-size heuristic on a Digital Annealer follows from Parizy and Togawa (2021), while Montañez-Barrera et al. (2024) propose a penalty dispensing with auxiliary variables altogether, reporting coefficients tuned on a thirteen-item problem. None of these evaluations involves minor embedding, chain integrity, or the finite dynamic range of physical couplers. The size exponent, on which the first two disagree by an order of magnitude, has never been independently re-estimated.

This thesis asks whether penalty calibration for the QKP, characterized almost entirely in simulation, survives the passage to physical hardware. The question divides along three axes, each turned into a falsification protocol: (P1) the dependence of the optimal penalty on instance size, (P2) the viability of the slack encoding under minor embedding, and (P3) the transfer of a calibrated penalty across instances, densities, and from a classical criterion to the annealer. P1 re-estimates the size exponent against a classical exact oracle and checks it on a QPU at three sizes. P2 measures what the slack encoding costs on two D-Wave topologies, where its auxiliary variables push the problem past the largest embeddable clique. P3 carries parameters calibrated on one instance, against a classical criterion, onto the other three targets.

The empirical target is the benchmark of Billionnet and Soutif (2004), restricted to its ten instances at $n = 100$ and density 25%, for which proven optima are published. Every one of the

ten admits a penalty setting under which the QPU returns a feasible solution with reliable chains, and eight reach the proven optimum exactly. What follows is an account of how that setting is found, of the three published rules that fail to predict it, and of one measurement decision that decides the sign of a conclusion.

Chapter 2 builds the problem, its classical resolution, the annealer and the QUBO encoding, ending on three gaps in the calibration literature. Chapter 3 sets out the instances, the pipeline, the hardware protocol and the calibration procedures, reporting the successive failures that led to the two-dimensional grid search retained at the end. Chapter 4 reports the three protocols. Chapter 5 interprets them and states the limitations, and Chapter 6 closes.

2 Literature Review

The Quadratic Knapsack Problem solved on D-Wave hardware sits between two traditions that do not speak the same language. Combinatorial optimization reasons in worst-case complexity and provable optima; quantum annealing reasons in energy landscapes, physical couplers, and finite precision. Penalty calibration is where they meet, and this chapter's recurring claim is that what the modelling literature treats as a parameter choice becomes a question about couplings once the computation leaves the simulator. Sections 2.1 to 2.4 build the problem, its classical resolution, the annealer, and the QUBO encoding that binds them; Section 2.5 turns to what the literature claims about transferring a calibrated penalty, and to the three gaps that claim leaves open.

2.1 The Quadratic Knapsack Problem

2.1.1 From the linear to the quadratic knapsack problem

In the classical 0–1 Knapsack Problem (KP01), studied since the 1950s by Dantzig (1957), n items each carry a profit p_i and a weight w_i , and a container of capacity C must be filled so as to maximize total profit without exceeding that capacity (Cacchiani et al., 2022; Martello and Toth, 1990). Writing $x_i \in \{0, 1\}$ for the decision to select item i , the model reads:

$$\max \quad \sum_{i=1}^n p_i x_i \quad (1)$$

$$\text{s.t.} \quad \sum_{i=1}^n w_i x_i \leq C, \quad x_i \in \{0, 1\} \quad (i = 1, \dots, n) \quad (2)$$

The objective (1) is additive, so the value contributed by a selected item is independent of the others. Karp (1972) placed the KP01 among the NP-hard (Nondeterministic Polynomial time-hard) problems, but its hardness is more precisely characterized as *weakly* NP-hard (Garey and Johnson, 1978; Pisinger, 2005), enabling dynamic programming (DP) in pseudo-polynomial time $O(nC)$ (Martello and Toth, 1990).

Economic applications rarely respect additivity. Projects exhibit dependencies that generate synergies or mutual exclusions, from the selection of risky interrelated investment portfolios (Laughunn, 1970) to the installation of freight handling terminals with shared fixed costs (Rhys, 1970), and no linear objective captures them (Beliakov, 2022; Galli et al., 2025). Gallo et al. (1980) accordingly formalized the QKP, adding to the objective a matrix of pairwise profits p_{ij} collected only when items i and j are both selected. The capacity constraint carries over unchanged, and the objective becomes a quadratic pseudo-Boolean function.

$$\max_{x \in \{0,1\}^n} f(x) = \sum_{i=1}^n p_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n p_{ij} x_i x_j \quad (3)$$

$$\text{s.t.} \quad \sum_{i=1}^n w_i x_i \leq C, \quad x_i \in \{0, 1\} \quad (i = 1, \dots, n) \quad (4)$$

The interaction profit p_{ij} is positive for a synergy and negative for a trade-off, although standard QKP formulations restrict both p_{ij} and p_i to non-negative values, which makes (3) superadditive and yields what Pisinger (2007) terms the *supermodular knapsack problem* (Beliakov, 2022; Galli et al., 2021).

The added term moves the problem into a different complexity class. Unlike the KP01, the QKP is strongly NP-hard, a property established by reduction from the strongly NP-complete Clique problem¹ (Gallo et al., 1980; Garey and Johnson, 1978; Pisinger, 2007). Strong NP-hardness rules out any pseudo-polynomial time algorithm unless $P = NP$, so the dynamic program that solves the KP01 in $O(nC)$ has no direct analogue here, and approximation offers no refuge either, the optimal profit resisting constant-factor approximation in polynomial time under standard complexity assumptions even with unit weights and under the non-negativity retained throughout this thesis (Fomeni & Letchford, 2014; Khot, 2004).

2.1.2 Compact formulation and instance parameters

Following Gallo et al. (1980), the QKP admits an equivalent matrix representation, adopted from this point forward. Let $V = (v_{ij})$ be the symmetric $n \times n$ non-negative matrix with $v_{ii} = p_i$ for all i and $v_{ij} = v_{ji} = p_{ij}/2$ for all $i \neq j$. Because $x_i^2 = x_i$ over binary variables, this construction folds the linear profits into the diagonal of V without loss of generality (Fomeni and Letchford, 2014), so that (3)–(4) reduces to

$$\max_{x \in \{0,1\}^n} x^\top V x \quad (5)$$

$$\text{s.t.} \quad \sum_{i=1}^n w_i x_i \leq C. \quad (6)$$

1. The reduction sets $w_i = 1$ and $p_i = 0$ for every item, $C = k$, and $p_{ij} = 1$ exactly when $\{i, j\}$ is an edge of the input graph; the QKP optimum then reaches $k(k-1)/2$ if and only if that graph contains a clique of size k .

In the benchmark family studied here, the entries of V are non-negative integers (Billionnet and Soutif, 2004).

Two instance parameters recur in every subsequent chapter. The density d gives the fraction of pairwise coefficients that are non-zero. One might expect d to govern the connectivity of the Quadratic Unconstrained Binary Optimization (QUBO) model derived in Section 2.4, and it does not, since the capacity penalty couples every pair of variables whatever the value of d (Appendix A.1.1). Density therefore governs the sparsity of the objective alone.

The tightness ratio $\alpha = C/\sum_i w_i$ measures how binding the capacity constraint is, and with it the fraction of $\{0, 1\}^n$ that remains feasible. Ohno et al. (2024) observe that tightness, long recognized as a determining factor for the multidimensional knapsack problem, had not previously been treated as a first-class instance feature in the QKP literature. Section 2.5 exploits this observation directly.

2.1.3 Reference instances

The empirical backbone of this work is the benchmark set of Billionnet and Soutif (2004). Under the generation protocol common to the earlier literature (Billionnet and Calmels, 1996; Caprara et al., 1999; Chaillou et al., 1989; Gallo et al., 1980; Hammer and Rader Jr., 1997), profit coefficients are integers drawn uniformly in $[0, 100]$, weights are integers drawn uniformly in $[1, 50]$, and the capacity is an integer selected between 50 and $\sum_i w_i$. The set spans sizes $n \in \{100, 200, 300\}$ and densities $d \in \{25, 50, 75, 100\}\%$, with proven, publicly available optima. Those optima, together with adoption by Parizy and Togawa (2021), Bontekoe et al. (2023) and Ohno et al. (2024), have made the set canonical and allow direct comparison with prior work. Ground truth throughout this thesis is taken from the published optima, with no independent re-solving.

Whether a QPU can accommodate them at all is uncertain, since the clique-size ceiling established in Section 2.3.4 may place a dense instance at $n \geq 100$ beyond reach; Section 3.1.2 documents the smaller-scale contingency set prepared against that outcome, generated from the correlation types of Pisinger (2005) and following the approach used elsewhere in this research group for related combinatorial problems on D-Wave hardware (Deleplanque et al., 2025).

The QKP thus retains the compact statement of the KP01 while inheriting none of its tractability. What classical methods achieve against that intractability, and at what cost, is the subject of Section 2.2.

2.2 Classical Resolution of the QKP

Casting a 0–1 quadratic problem as a Mixed-Integer Linear Program (MILP) conventionally relies on the linearization of Fortet (1960), which substitutes an auxiliary variable $y_{ij} = x_i x_j$ for each interacting pair, enforced by $y_{ij} \leq x_i$, $y_{ij} \leq x_j$ and $y_{ij} \geq x_i + x_j - 1$. The decision space expands from n binary variables to $O(n^2)$, with as many added inequalities. The QUBO

encoding of Section 2.4 avoids that expansion in the objective and transfers it to the capacity constraint through slack discretization.

Exact frameworks rely on branch-and-bound driven by tight relaxation bounds. Caprara et al. (1999) derive upper bounds from a Lagrangian relaxation of the linearization constraints and solve instances at full density up to $n = 400$, a ceiling that drops to $n \leq 120$ once density falls to 25%. Their profile inverts the conventional ordering, sparse instances resisting exact resolution more stubbornly than dense ones, and every instance evaluated in this thesis sits at that sparse end. The Lagrangian decomposition of Billionnet and Soutif (2004) clears low- and medium-density instances to $n = 300$, taking 4 minutes at $n = 100$ and 45 minutes at $n = 150$, against earlier schemes that stalled near $n = 100$ (Hammer & Rader Jr., 1997). Semidefinite relaxations are theoretically tighter and remain impractical on memory and runtime (Galli et al., 2025; Helmberg, 2000), while heuristic frameworks reach to $n = 6000$ (Chen & Hao, 2017), beyond the range evaluated here.

Dynamic programming for the KP01 tracks a compact state $[i, c]$ over the item index and the consumed capacity, a representation the additive objective (1) makes sufficient. Under the quadratic objective (5) that sufficiency collapses, the marginal profit of item i depending on every previously selected item through the coefficients v_{ij} , so an exact state must carry the complete subset and the recursion degenerates into 2^n enumeration. Classical DP on the QKP accordingly resorts to heuristic state-space pruning (Fennich et al., 2024; Fomeni & Letchford, 2014) or to topology-restricted exact schemes (Rader Jr. & Woeginger, 2002). Encoding subsets as n -bit masks and evaluating objective and constraint across all 2^n configurations costs $O(2^n \cdot n^2)$, which standard CPUs handle to $n \approx 25$; Section 3.1.3 adopts that enumeration as the ground-truth oracle for $n \leq 20$, the published optima of Billionnet and Soutif (2004) supplying exact values at $n = 100$.

No classical exact route preserves the quadratic structure inside the solver, each paying either $O(n^2)$ variables in linearization or 2^n states in enumeration. Bypassing that overhead is what a hardware-native encoding offers.

2.3 Quantum resolution of the QKP

2.3.1 Gate-based search and its current limits

Two models of quantum computation address optimization. The gate model applies circuits of unitary operations to qubits within a general-purpose framework, its emblematic optimization primitive being Grover's search, which identifies a target entry among N in $\pi/4 \cdot \sqrt{N}$ iterations, a quadratic speedup provably optimal for unstructured search (Bennett et al., 1997; Grover, 1996). Over $\{0, 1\}^n$ the search space holds $N = 2^n$ configurations, where the Grover-based procedures of Creemers and Armas (2025) match state-of-the-art classical exact bounds on the linear KP01 and outperform them on the quadratic variant, presupposing throughout a fault-tolerant universal quantum computer at scale that no hardware roadmap places within near-term reach

(Creemers, 2025; Creemers & Armas, 2025; Preskill, 2018). Quantum annealing instead evolves a time-dependent Hamiltonian toward the ground state of an energy function (Farhi et al., 2000; Kadowaki & Nishimori, 1998), confining operation to minimization while matching the QKP exactly, as Section 2.3.3 develops. Grover-based optimization thus remains pertinent in theory and inaccessible in practice, whereas physical annealing hardware exists today at the scale of the evaluated instances.

2.3.2 *The physics of Quantum Annealing*

Quantum annealing initializes a system in the trivial ground state of a transverse mixing Hamiltonian and interpolates toward a problem Hamiltonian whose eigenstates encode classical binary configurations,

$$H(t) = A(t)H_{\text{initial}} + B(t)H_{\text{problem}}, \quad (7)$$

where $A(t)$ decreases monotonically to zero while $B(t)$ increases over a microsecond-scale schedule (Kadowaki and Nishimori, 1998). Under a sufficiently slow evolution, the adiabatic theorem guarantees that the system remains in its instantaneous ground state, yielding the global minimum of H_{problem} upon final measurement (Farhi et al., 2000). Over the discrete hypercube $\{0, 1\}^n$, the QUBO energy landscape is traversed by quantum tunneling through narrow potential barriers, where a classical sampler would need thermal activation to pass over them (Kadowaki and Nishimori, 1998).

Physical QPUs operating in the NISQ era depart systematically from the adiabatic ideal. Finite annealing schedules, thermal fluctuations, and spectral degeneracies reduce the theoretical exact solver to a probabilistic sampling heuristic (Yarkoni et al., 2022), while the minimum spectral gap governing the required schedule duration resists general analytical estimation (Farhi et al., 2000). In the context of knapsack optimization, Pusey-Nazzaro and Date (2020) provide an early cautionary benchmark. Evaluating small linear knapsack instances on a D-Wave 2000Q, whose Chimera topology offers a per-qubit degree of six, they recover the exact optimum on none of their test cases and concede that the cause cannot be probed directly. Their reading charges adiabatic optimization itself, on hardware two generations behind the systems Section 2.3.4 describes.

2.3.3 *Structural alignment between the QKP and Quantum Annealing*

The QKP exhibits a native isomorphism with quantum annealing architectures. Direct mapping of the quadratic binary objective onto a QUBO matrix requires no auxiliary variable expansions or linearization gadgets, allowing the mathematical formulation to coincide with the physical Ising Hamiltonian (Lucas, 2014). The penalty calibration rules evaluated here are themselves formulated as QUBO parameters, previously studied on classical simulators and digital Ising machines (Bontekoe et al., 2023; Montañez-Barrera et al., 2024; Ohno et al., 2024).

Testing these laws at $n = 100$ variables requires hardware that executes binary quadratic forms natively, and only the annealer does so at that scale.

2.3.4 The hardware processor topologies: Advantage and Advantage2

The empirical evaluations in this thesis compare two generations of D-Wave quantum annealers. Due to physical fabrication variances, a fraction of qubits and couplers remain disabled, defining the user-accessible *working graph* (“Topologies — D-Wave Quantum Computing Products Documentation”, n.d.). The Advantage system, built on the Pegasus topology (Boothby et al., 2019), is fabricated with 5 640 qubits, yielding 5 616 operational qubits at a per-qubit connectivity degree of $k = 15$ on the active working graph (Deleplanque et al., 2025; “Topologies — D-Wave Quantum Computing Products Documentation”, n.d.). The Advantage2 processor transitions to the Zephyr topology (Boothby et al., 2021), increasing per-qubit degree to $k = 20$. While a full-scale Z_{15} fabric accommodates 7 440 qubits, the physical Advantage2 QPU evaluated here provides 4 577 functional qubits.

Raw qubit count does not constitute a valid performance metric for quadratic knapsack formulations. The capacity penalty derived in Section 2.4 couples every logical variable pair across all problem densities (Appendix A.1.1), strictly bounding the tractable problem size by the maximum embeddable complete graph (K_L). On the active working graphs, the deterministic buscliq compiler constructs maximum cliques of K_{177} on Pegasus and K_{106} on Zephyr. The higher connectivity degree of the Zephyr architecture does not offset its smaller qubit yield, imposing a more restrictive embedding ceiling on the newer processor generation. Chapter 4 quantifies the operational impact of this topological restriction at $n = 100$.

2.3.5 Embedding, chain integrity, and coupler precision

This thesis reserves *logical variable* for the binary decision variable x_i of the QUBO, and *physical qubit* for a single element of the Pegasus or Zephyr working graph, restricted to degree-15 or degree-20 local connectivity (Section 2.3.4). The gate-model concept of a *logical qubit*—a fault-tolerant module assembled from multiple physical qubits via error correction (Section 2.3.1)—has no physical counterpart in quantum annealing and is not used here.

Every logical variable of the QKP must interact with more variables than one physical qubit can reach, whatever the objective’s density, since the capacity penalty couples every pair (Appendix A.1.1). Minor embedding resolves the mismatch by mapping each logical variable to a *chain*, a set of physical qubits coupled ferromagnetically at strength J_F to enforce a uniform collective state (Venturelli and Kondratyev, 2019, Eq. 5; Appendix A.1.2). The magnitude of J_F competes directly with the problem’s own logical coupling scale (“Reference Documentation — D-Wave Quantum Computing Products Documentation”, n.d.): an under-calibrated J_F allows local field fluctuations to overcome the intra-chain alignment barrier, inducing internal chain disagreement where spin flips yield lower overall energy configurations. With problem scale n

forcing physical chain elongation, logical variable fidelity degrades (Boothby et al., 2019), so chain fragility follows instance size and not the density of the objective matrix.

The Chain-Break Fraction (CBF), reported natively by Ocean embedding composites (“Reference Documentation — D-Wave Quantum Computing Products Documentation”, n.d.), quantifies the proportion of broken chains per sample read. A broken chain leaves its logical variable without a deterministic binary value, establishing CBF as the primary reliability metric throughout this research. The literature treats CBF as a continuous quality signal governed jointly by chain strength, embedding topology, and problem scale, with no universal rejection threshold (Grant et al., 2021; Jeong et al., 2025; Pelofske, 2023).

In parallel, physical couplers operate within a finite analog dynamic range, so objective coefficients, penalty parameters, and chain strengths share a bounded energy budget. Coefficients programmed outside the chip’s specified range trigger a global rescaling of the problem (Rosenberg et al., 2016), which Ocean’s `auto_scale` (“QPU Solver Parameters”, n.d.) does not relieve. Uniform rescaling preserves every ratio it touches, while the machine’s ability to separate nearby energy levels stays fixed, governed by analog control errors and intrinsic noise (O’Gorman et al., 2015). A large penalty-to-objective or chain-to-objective ratio therefore compresses the objective toward a noise floor historically estimated at 2 to 4% of full scale, and the sampler loses the resolution to rank good feasible states against bad ones (Buonaiuto et al., 2023; Güney et al., 2025). Raising J_F well above the native couplings carries a second cost, stiffening the qubits against the state transitions the annealing dynamics depend on (Deleplanque et al., 2025; Pérez Armas et al., 2024).

These three physical constraints define the operational regime in which all penalty calibration experiments reported here take place. Chapter 4 quantifies the exact spatial distance between theoretical penalty parameters derived on classical exact oracles and the viable execution domains required on physical QPU hardware.

2.4 QUBO formulation of the QKP and the calibration problem

2.4.1 From a constrained problem to an unconstrained one

A quantum annealer minimizes an energy and recognizes no constraint, so the capacity limit must be absorbed into the objective as a penalty term (Lucas, 2014; Yarkoni et al., 2022). Writing $f(x)$ for the QKP objective (3),

$$H(x) = -f(x) + P(x), \tag{8}$$

where the minus sign turns the maximization into the minimization the hardware performs, and $P(x)$ is positive on every violation and zero otherwise. The admissible shape of P is narrower than that condition suggests. A QUBO carries linear and pairwise terms and nothing beyond, any higher-order construction requiring reduction to quadratic form at the cost of auxiliary variables

(Lucas, 2014). The two penalties compared in this chapter answer that ceiling in different ways, one by squaring a linear form and the other by truncating an expansion at second order. Equation (8) leaves open which to adopt, and with what coefficients.

2.4.2 The difficulty of an inequality constraint

A penalty must vanish on feasible configurations and rise on infeasible ones. For an equality constraint the squared deviation does exactly that. An inequality does not, since squaring $\sum_i w_i x_i - C$ charges every solution leaving capacity unused, and those solutions sit on the feasible side of (6). Two routes around that asymmetry organize the remainder of this thesis. The slack strategy pads the inequality into an equality with auxiliary binary variables and squares the result. The slack-free strategy of Section 2.4.4 builds the asymmetry into the penalty itself, charging violation more heavily than compliance. They diverge in calibration regime and in hardware profile, and the pivot instance `jeu_100_25_1` ($n = 100$, $d = 25\%$, $\alpha = 0.259$, $C = 669$) carries the comparison through the three that follow.

2.4.3 The slack formulation

The canonical construction (Lucas, 2014) introduces a slack quantity S with $0 \leq S \leq C$, turning the inequality into $\sum_i w_i x_i + S = C$, and encodes S over $\lceil \log_2(C + 1) \rceil$ auxiliary bits s_k with coefficients c_k , a bounded-integer-to-binary expansion traced back to Watters (1967). The squared deviation

$$P_{\text{slack}}(x, s) = \lambda \left(\sum_i w_i x_i + \sum_k c_k s_k - C \right)^2 \quad (9)$$

then vanishes on the feasible set and nowhere else. The expansion truncates its last coefficient so that the bits sum exactly to C , giving $c_k = 1, 2, \dots, 2^{k-1}, C - (2^k - 1)$; the pivot instance, at $C = 669$, requires ten bits carrying 1 through 256 and a final 158, so a 100-item problem occupies 110 logical variables.

Expanding the square produces cross terms $w_i w_j x_i x_j$ over every pair of items (Appendix A.1.1) and $w_i c_k x_i s_k$ between every item and every slack bit, extending the clique across all 110 variables, ten beyond the 100 the problem states. Weights reaching 50 against a largest coefficient of 256 put the raw item-slack terms $2w_i c_k$ at 2.6×10^4 , which λ lifts to 10^5 to 10^6 over the range Chapter 4 sweeps, against objective coefficients bounded by 100.

2.4.4 The slack-free formulation: unbalanced penalization

Montañez-Barrera et al. (2024) charge the violation directly, weighting infeasibility more heavily than compliance and introducing no auxiliary variable. Let $h(x) = C - \sum_i w_i x_i$ denote the constraint slack, feasibility corresponding to $h(x) \geq 0$. Their construction borrows its shape from the exponential decay $e^{-h(x)}$, near zero wherever capacity remains and growing sharply

once it is exceeded. Since that curve is not a QUBO (Section 2.4.1), they expand it to quadratic order,

$$e^{-h(x)} \approx 1 - h(x) + \frac{1}{2}h(x)^2, \quad (10)$$

free the two coefficients for tuning, and drop the constant, which displaces the energy without moving its minimum, leaving

$$P_{\text{unb}}(x) = -\lambda_1 h(x) + \lambda_2 h(x)^2. \quad (11)$$

The parabola has its vertex at $h^* = \lambda_1/(2\lambda_2)$, so the penalty on its own rewards leaving h^* of the capacity unused while the objective pulls the other way, and calibration decides where the two settle. What changes downstream is the variable count and the conditioning. Nothing auxiliary enters, so the pivot instance occupies 100 logical variables against the slack encoding's 110, and λ_2 is no longer pinned to the value a single squared term such as (9) imposes.

The authors state what the encoding costs. It does not guarantee that the optimal solution is the ground state of the cost Hamiltonian, only that it sits close to one (Montañez-Barrera et al., 2024). Their reported knapsack coefficients, $(\lambda_1, \lambda_2) \approx (0.96, 0.037)$, place the quadratic term more than an order of magnitude below the linear one in the units they report. Section 3.2.2 shows that the ordering reverses under the normalization adopted here, and Chapter 4 takes both to hardware.

2.4.5 Choosing the slack-free formulation

The two encodings part company on the axis the hardware cares about. At $n = 100$ the slack penalty (9) spreads its clique over 110 variables against a largest embeddable clique of K_{106} on Zephyr (Section 2.3.4), leaving clique-aware embedding nowhere to place it, while the slack-free penalty (11) stays at 100 and fits. Protocol P2 in Chapter 4 measures that margin and what it costs to close it by other means. The calibration work of this thesis runs on (11) throughout, the slack encoding reappearing only as the object of that protocol.

2.4.6 The calibration problem

Fixing the form of the penalty leaves its coefficients open, λ in (9) and (λ_1, λ_2) in (11), and the literature is clear on how they fail at either extreme. Too small, the objective dominates and the QUBO optima violate the constraint. Too large, the penalty flattens the landscape until the sampler can no longer separate good feasible states from bad ones. Bontekoe et al. (2023) put numbers on that tension for the QKP. Enforcing the constraint in theory calls for penalties of order 10^4 on their instances, magnitudes that swamp the QUBO entries and degrade annealing so badly that they work inside a window from 1 to 20 instead, its upper end taken from the $\lambda = \text{strength} \cdot n \cdot d$ heuristic of Parizy and Togawa (2021). Neither bound anticipates the coupler precision budget of Section 2.3.5, which narrows the window further on physical hardware.

Setting a penalty and predicting one are different problems. Finding the value that works on a given instance is a search, run by grid evaluation or by the Nelder-Mead simplex (Nelder and Mead, 1965) that Montañez-Barrera et al. (2024) apply to their own multipliers. Establishing a rule $\lambda(n, d, \alpha)$ that holds without re-calibration is a claim of another order, and Section 2.5 reviews what has been attempted.

2.5 State of the art in parameter transfer

2.5.1 The near-invariant scaling law of Ohno et al. (2024)

Ohno et al. (2024) propose the first analytical penalty rule for the QKP meant to spare per-instance tuning. Working in the slack formulation on the instances of Billionnet & Soutif, they model the optimal coefficient as a product of instance features,

$$\lambda_{\text{estimate}} = A \cdot n^{c_n} \cdot d^{c_d} \cdot \alpha^{c_\alpha}, \quad (12)$$

and fit it by log-linear regression on the penalties that prove optimal under simulated annealing over 100 instances, obtaining $A = 1.14$, $c_n = 0.09$, $c_d = 0.84$ and $c_\alpha = -0.21$ (Ohno et al., 2024). What that regression takes as its response variable determines what the exponent describes. Their $\lambda_{\text{SA-RI}}$ is the coefficient optimal for simulated annealing *combined with a two-stage post-processing*, a repair procedure that restores feasibility to infeasible samples and an improvement procedure that refines feasible ones, selected by lexicographic order over best objective, success rate and mean objective on a dyadic grid whose lower end is 2^{-6} , with instances solvable to optimality by a greedy heuristic excluded because their optimal coefficient is trivially zero (Ohno et al., 2024). Their own account separates two quantities on that basis. The critical coefficient $\lambda_{\text{critical}}$ marks where feasibility begins and depends heavily on the encoding of the inequality; the optimal coefficient they regress lies below it, the repair procedure supplying feasibility that the penalty no longer has to enforce, and is correspondingly robust to that encoding (Ohno et al., 2024).

The exponent on n is where the interest lies. At $c_n = 0.09$ the optimal penalty barely moves with instance size, a weak dependence the authors report having anticipated, and their large-scale runs on the Fixstars Amplify Annealing Engine, a GPU-based Ising machine, drop n from the rescaling altogether in favour of $\lambda = a \cdot (d/100) \cdot \sqrt{1/\alpha}$ with integer a .

Chapter 4 re-estimates the size exponent at 0.97 on a calibration campaign that involves no hardware, an order of magnitude above their figure. The two estimates do not share a response variable, and Section 5.1 takes up what follows from that. What holds whatever the resolution is that the rule has never met a quantum processor. Neither simulated annealing nor the Amplify engine embeds anything, so no chain ever competes with the objective for coupler precision (Section 2.3.5), and neither carries the repair layer that a QUBO submitted directly to a QPU does without. The authors add a caveat of their own worth recording, that the fit rests on one

instance-generation procedure and need not hold under another.

2.5.2 *The linear-in-size heuristic of Parizy and Togawa (2021)*

A second line of work posits the opposite dependence on instance size. Parizy and Togawa (2021) solve the QKP on a Digital Annealer, a distinct Ising machine, using the slack formulation of Glover et al. (2002) on the same Billionnet & Soutif benchmark. Their penalty coefficient, written λ here to keep it apart from the tightness ratio, is set to $\text{strength} \cdot n \cdot d$ with strength starting at 0.1, on the reasoning that a larger and denser problem yields more potential profit per added item and so demands a proportionally stronger penalty. Nothing is fitted, since strength simply rises by 50% and the run repeats whenever every replica lands on an infeasible solution. The functional form nonetheless implies $c_n = 1$ in the notation of (12), against the $c_n = 0.09$ of Ohno et al. (2024), and the two published positions cannot both hold.

The same paper reports the dynamic-range tension of Section 2.3.5 on hardware of another make, constraint Hamiltonian values running an order of magnitude above profit values, and answers by holding the penalty deliberately low and repairing infeasible solutions in classical post-processing, which places it alongside Ohno et al. (2024) among the rules fitted with a repair layer in place.

2.5.3 *The slack-free penalty of Montañez-Barrera et al. (2024)*

The penalty (11) was proposed to spare the qubits and couplings that slack variables consume (Montañez-Barrera et al., 2024). Its evaluation spans QAOA on gate-model simulators and quantum annealing, over the travelling-salesman, bin-packing, and knapsack problems, with the QKP absent from the test set. Physical D-Wave Advantage hardware enters for bin packing alone, where the method reaches instances the slack approach cannot, and the Zephyr generation not at all.

Their knapsack coefficients carry a narrower warrant than a scaling law does. Nelder-Mead fixed $(\lambda_1, \lambda_2) \approx (0.96, 0.037)$ on a single 13-item problem, after which the pair was checked against other random cases, and the authors note that parameters tuned on one distribution need not carry to another (Montañez-Barrera et al., 2024). Whether that pair survives the step to $n = 100$ on the Billionnet & Soutif instances, and how ferromagnetic chains behave under a penalty this asymmetric, are both untested.

2.5.4 *What simulation and hybrid solvers leave open*

Bontekoe et al. (2023) rank six QUBO encodings of the capacity constraint under simulated annealing, and the ranking repays attention. The standard binary-slack encoding, their Type 1, sits among the weakest, while the best performer is their Type 5, alone among the six in carrying no auxiliary binary variable. Additional variables enlarge the energy landscape and make good solutions harder to reach, runs their explanation. They are equally clear about what Type 5 costs,

since the offset it substitutes for those variables has to be gauged for each use case. Trading a variable count for a calibration problem is precisely the bargain this thesis inherits, (11) buying its 100 variables at the price of two coefficients nobody knows how to set in advance. Their result also hardens when read against hardware. Dropping auxiliary variables already pays in simulation, where no topology constrains anything; at $n = 100$ on Zephyr it decides whether clique-aware embedding has anywhere to place the problem at all.

Penalty characterization for the multi-knapsack problem keeps the question live (Güney et al., 2025), and keeps it analytical. Hybrid solvers evade it from the other side. Asawalertsak et al. (2025) compare D-Wave's BQM and CQM routes, the latter taking the inequality as a constraint and delegating it to a classical layer, so that no penalty coefficient is ever exposed to the annealer. Studying calibration requires the QUBO to reach the hardware intact, which is why this thesis samples the QPU directly and not through a hybrid solver. The earliest campaigns that did so had no rule to apply and tuned instance by instance (Rosenberg et al., 2016; Venturelli and Kondratyev, 2019).

2.6 What the survey leaves unsettled

The QKP is strongly NP-hard, and no classical route preserves its quadratic structure inside the solver, which is what makes a native QUBO encoding worth its own difficulties. Those difficulties reduce to a penalty coefficient, and the physical regime of Section 2.3.5 decides which values are available at all. Three gaps separate what the literature has settled from what a calibration campaign on hardware would need.

The size exponent of (12) rests on a single measurement. Ohno et al. (2024) report $c_n = 0.09$ from one regression on one instance generator, evaluated where no embedding takes place, Parizy and Togawa (2021) proceed from $c_n = 1$ without fitting anything, and neither figure has been checked against a physical processor. Nor has the slack encoding been characterized at $n = 100$ on a QPU, Bontekoe et al. (2023) testing in simulation, where no topology constrains anything, and the Amplify engine having no chains to break (Ohno et al., 2024). The third gap goes unrecognized as one, since the calibration literature assumes throughout that a penalty optimal against a classical criterion is optimal on hardware.

Chapter 4 supplies the missing re-estimate, characterizes the slack encoding on both topologies, and answers the third only directionally, within the experimental scope Section 3.7 sets out.

3 Methodology

Three claims from Chapter 2 set the terms of the empirical work. The scaling law of Ohno et al. (2024) holds the optimal penalty almost invariant to instance size, at $c_n \approx 0.09$. The unbalanced formulation of Montañez-Barrera et al. (2024) carries parameters tuned on one problem over to different and larger ones. Beneath both lies an assumption the QKP calibration

literature has had no occasion to state, that a penalty established against a classical or simulated criterion is the penalty a physical QPU needs.

Turning these into an experimental programme raises two operational questions. How many Billionnet-scale instances can be solved on current QPU hardware, and how many QUBO formulations can be made to fit? The instance sets, the formulation, and the protocols of Section 3.6 all follow from the answers.

The calibration procedure itself was not reached directly, and the chapter reports the attempts in the order they were made, each failure being the argument for the next. Two-dimensional Nelder-Mead calibration ran into a flat valley in (κ_1, κ_2) and had to be canonicalized down to one dimension. The resulting log-linear regression then failed to transfer to Billionnet-scale instances. An empirical anchoring with a tightness correction followed, and gave way on precisely the instances that needed it most. The two-dimensional grid search retained at the end survives those three limitations.

3.1 Instances and ground truth

3.1.1 The Billionnet published benchmark

The empirical target is the benchmark of Billionnet and Soutif (2004) described in Section 2.1.3, restricted to the ten $n = 100$ instances at density $d = 25\%$, the `jeu_100_25_*` series. The instance `jeu_100_25_1` ($\alpha = 0.259$, $C = 669$) anchors every initial calibration decision, its tightness sitting near the centre of the range the ten instances span ($\alpha \in [0.063, 0.934]$). Ground truth comes from the published optima, with no independent re-solving.

3.1.2 Generated calibration sets

Calibration requires instances small enough for the exact oracle of Section 3.1.3, which a benchmark starting at $n = 100$ cannot supply. Two protocols generated the sets that follow.

The *exact* campaign uses a synthetic generator over $n \in \{10, 20\}$, densities $d \in \{0.25, 0.5, 0.75, 1.0\}$, one hundred values of $\alpha \in [0.01, 1.0]$ and ten seeds per cell, for 8 000 instances. It ran twice over the identical grid, once with κ_2 left to the optimizer and once with κ_2 fixed at 0.1, and Section 3.4.1 uses the pair to establish the canonical point calibration requires.

The *Billionnet* campaign follows the published protocol of Section 2.1.3 and departs from it on the capacity, set here to $C = \max(50, \lfloor \alpha \sum_i w_i \rfloor)$ instead of drawn at random, since a regression needs α as a controlled variable and not as a by-product. Eighteen values of α replace the hundred, evenly spaced from 0.10 to 0.95, a capacity near zero admitting almost no item and a capacity near $\sum_i w_i$ admitting them all, neither end leaving the penalty anything to arbitrate. Its 1 440 instances, calibrated at $\kappa_2 = 0.25$, supply the observations the regression of Section 3.4.2 is fitted on.

Three differences separate the two generators, and Appendix A.3 states both in full. The synthetic generator fills every linear profit whatever the density, while the Billionnet protocol

subjects the diagonal to d along with the rest, leaving a fraction of items with no individual profit at all. Weights run to 100 against the published protocol's 50, and the capacity floor sits at $\max_i w_i$ instead of at 50. Section 3.4.2 measures what this costs in transfer.

A third set was generated and never used. Two hundred and seventy instances at $n \in \{10, 20, \dots, 90\}$, drawn from the correlation types of Pisinger (2005), stood ready in case no Billionnet-scale instance proved embeddable, a contingency the slack-free formulation removed by fitting at $n = 100$ on Zephyr. They remain available for later work.

3.1.3 Ground truth by regime

Two oracles cover the two instance regimes. At $n = 100$ ground truth is the published Billionnet & Soutif optimum (Section 3.1.1). At $n \leq 20$ the bitmask enumeration of Section 2.2 supplies it for the classical calibration campaigns, evaluating objective and constraint over all 2^n configurations in vectorized form, multicore.

Enumeration remains tractable to roughly $n = 25$, and the bound is set at 20 deliberately. The oracle runs inside the Nelder-Mead loop of Section 3.4.1, not once per instance, and the exact campaign alone spends a median of 153 objective evaluations on each of its 8 000 instances, some 1.2 million exhaustive enumerations in total. The product of the two, rather than any single evaluation, fixes the ceiling. Nothing bridges the two regimes here, which Section 3.7 records as the prerequisite for any work in between.

3.2 QUBO construction and normalization

3.2.1 From the instance to the BQM

The objective coefficients V are built as in Section 2.1.2 and divided by $P_{\max} = \max_{ij} |v_{ij}|$, which places every objective coupler in $[-1, 1]$. The capacity constraint enters through Ocean's `add_linear_inequality_constraint` primitive under method `unbalanced`, for the reason Section 2.4.5 gives, its penalty parameterized in units of $W_{\max} = \max_i w_i$.

In raw units a penalty coefficient of 10 buys a different constraint-to-objective tension on every instance, the tension depending on how large that instance's weights and profits happen to be, and no calibrated value means anything outside the instance it was fitted on. Divided through, the same coefficient buys the same tension everywhere, a precondition for asking whether one instance's calibration transfers to another at all.

3.2.2 Physical and dimensionless penalties

Two parameters describe the same penalty at different points in the pipeline, and any transfer claim rests on holding them apart. The *physical* λ carries the units of the raw problem, so instances whose weights differ in magnitude need different λ to reach the same effective constraint strength. The *dimensionless* (κ_1, κ_2) carry the pipeline's arithmetic, the objective having been

divided by $P_{\max}$ and the penalty expressed in units of $W_{\max}$. Preserving the energy of (11) through both divisions fixes the conversion,

$$\kappa_1 = \lambda_1 \frac{W_{\max}}{P_{\max}}, \quad \kappa_2 = \lambda_2 \frac{W_{\max}^2}{P_{\max}}, \quad (13)$$

which on the pivot instance ($W_{\max} = 50$, $P_{\max} = 100$) reads $\kappa_1 = 0.5 \lambda_1$ and $\kappa_2 = 25 \lambda_2$. Writing $\tilde{h}(x) = h(x)/W_{\max}$ for the slack in the same units as the penalty carries (11) through both divisions unchanged. The conversion already separates this thesis from the reference it starts out from. The knapsack coefficients of Montañez-Barrera et al. (2024), $(\lambda_1, \lambda_2) \approx (0.96, 0.037)$, place the quadratic term more than an order of magnitude below the linear one in physical units, where the same pair reads $(\kappa_1, \kappa_2) \approx (0.48, 0.93)$ once carried through (13), the quadratic coefficient undergoing the division by $W_{\max}$ twice and overtaking the linear one. Their ratio κ_1/κ_2 falls from 26 to 0.52, against the 52 that Chapter 4 calibrates at $n = 100$, a factor 100 and a reversal of the two coefficients' order. Every calibrated value reported below lives in κ space, and (13) carries it back whenever a physical coefficient is wanted.

3.3 Hardware protocol

Every QPU campaign runs on the Advantage2 system and its Zephyr topology, whose clique ceiling (Section 2.3.4) is the tighter of the two and therefore decides the choice of formulation instead of merely favouring it. Zephyr's per-qubit connectivity of 20, against Pegasus's 15, suits a QUBO that is complete whatever the objective's density. Pegasus enters only through the exploratory campaign of Section 4.3, run to separate what is topological in the slack obstruction from what is not. No systematic study was carried out there.

A single clique embedding is computed once per instance with `busclique` and reused across every sample through `FixedEmbeddingComposite`. Recomputing it at each call would let the physical qubits and couplers change from one grid cell to the next, leaving differences in the sampled distribution no longer attributable to (κ_1, κ_2) alone.

Chain strength stays at Ocean's `uniform_torque_compensation` (UTC) default with prefactor 1.414 ("QPU Solver Parameters", n.d.), which scales it as $\sqrt{\text{connectivity}} \times \text{coupling}_{\text{RMS}}$. Tuning it per cell would improve chain integrity (Park and Lee, 2024) at the cost of the measurement, given that a chain-break fraction observed downstream could then be charged to the chain-strength policy as readily as to the penalty. Held fixed, it leaves cbf a function of (κ_1, κ_2) and of nothing else.

3.3.1 Sampling

Each call to the QPU draws `num_reads = 2000` samples at the annealing time the solver exposes by default, and every parameter cell is submitted ten times over. Splitting the samples across ten programming cycles instead of drawing 20 000 in a single call makes the repetition

informative, the analog control errors of Section 2.3.5 holding fixed for the duration of one programming and only a fresh one resampling them (O’Gorman et al., 2015). A cell therefore carries run-to-run variability and not sampling noise alone, and Section 3.5 states which statistic of the submissions each metric reports.

Broken chains are resolved by the majority vote Ocean applies by default, which assigns each logical variable the value held by most qubits of its chain. Every sample is read after that resolution, and both the weight and the objective are recomputed classically from the resulting selection rather than taken from the energy the sampler returns, so no reported figure depends on the machine’s own accounting. The operation restores an assignment; it does not restore feasibility, a majority-voted sample violating the capacity constraint as readily as the raw one. No repair procedure of the kind Ohno et al. (2024) apply downstream of their sampler is used anywhere in this thesis, so the penalty enforces the constraint unaided, and the chain-break fraction of Section 3.5 records how much of the assignment the vote had to reconstruct.

Classical calibration runs multicore, QPU campaigns sequentially against a single solver. Appendix A.4 records the memory and checkpointing measures the campaigns required, none of which bears on the results.

3.4 Calibration procedures

3.4.1 Nelder-Mead calibration with fixed κ_2

The unbalanced penalty (11) leaves two parameters free, and the set of pairs that perform comparably at fixed instance parameters does not form a unique minimum. Güney et al. (2025) establish the analytical form of this degeneracy for the multi-knapsack problem, where every multiplier above a computable bound reformulates the problem exactly; the campaigns reported here identify a similar empirical indeterminacy for the QKP across both coefficients simultaneously. When applied to both parameters, the Nelder-Mead algorithm therefore converges to a local solution its initialization dictates within a flat subspace, and calibrated values originating from different (n, d, α) configurations become incomparable. The regression of Section 3.4.2 measures what that costs. Fitted on a campaign that leaves κ_2 to the optimizer it returns $R^2 = 0.11$; fitted on a campaign of identical design with κ_2 fixed, $R^2 = 0.90$. Without a canonical reference κ_1 is no longer a reliable function of the instance parameters.

Two methods of imposing such a reference were evaluated on a reduced grid of eight cells ($n \in \{10, 15, 18, 20\}$, $d \in \{0.25, 0.5\}$, $\alpha = 0.26$, ten seeds per cell). Applying a Tikhonov regularization term $\rho \|\kappa\|^2$ (Tikhonov, 1963) shapes the valley into a well-defined minimum, while fixing κ_2 and optimizing κ_1 alone removes the degeneracy by reducing the dimensionality of the search. Both approaches return κ_1 values within 10% of each other across every cell, and the fixed- κ_2 strategy yields a tighter seed-to-seed coefficient of variation in seven of the eight cells, which decided the choice. Leaving κ_2 to the optimizer also allows solutions to approach zero and eliminate the penalty’s quadratic component, while $\kappa_2 = 0.1$ holds roughly

one and a half decades of separation from κ_1 . The Billionnet campaign detailed in Section 3.1.2 subsequently increases this parameter to $\kappa_2 = 0.25$, aligning with the optimal value calibrated empirically on the hardware for the pivot instance.

The objective is

$$\text{loss}(\kappa_1) = \text{gap_energy}(\kappa_1, \kappa_2) + \rho \kappa_1^2, \quad \rho = 0.1, \quad (14)$$

with gap_energy the QUBO energy at the proven optimum minus the minimum QUBO energy over $\{0, 1\}^n$, evaluated by the oracle of Section 3.1.3. A zero would place the proven optimum at the ground state of the QUBO, and it is reached on 22.5% of the campaign. Its rarity belongs to the encoding and not to the search, unbalanced penalization guaranteeing only that the optimum lies near the ground state, never that it coincides with it (Montañez-Barrera et al., 2024). The calibration therefore minimizes a distance the formulation does not allow to vanish.

The regularization term accounts for most of the objective it is added to, taking between 74% and 77% of the total loss across the eight cells of the Billionnet campaign and leaving the calibrated κ_1 below the value that would minimize the gap alone. The displacement can be measured exactly, the QUBO energy being affine in κ_1 , so that the smallest minimizer of gap_energy , the $\rho \rightarrow 0$ limit, follows in one pass over the 2^n states. Refitting (15) on those unregularized values moves every exponent by 3 to 11% and leaves the fit unchanged at $R^2 = 0.803$, c_n rising to 1.004 from 0.972. The bias therefore flattens the dependence it is applied to, most where κ_1 is largest, without displacing the conclusion of Section 3.6.1.

Nelder-Mead runs from three initializations spanning the κ_1 axis ($\kappa_1 \in \{0.1, 5.0, 15.0\}$), with `maxiter` at 300 and tolerances of 10^{-4} on the parameter and 10^{-6} on the objective, keeping the best of the three.

3.4.2 Regression and the failed transfer to Billionnet-scale instances

Following Ohno et al. (2024), a log-linear regression is fitted by ordinary least squares on the calibrated κ_1^* ,

$$\log \kappa_1^* = \log A + c_n \log n + c_d \log d + c_\alpha \log \alpha + \varepsilon. \quad (15)$$

Two campaigns, the second closer to the target instances than the first, produce successive versions of it.

The first is fitted on the synthetic campaign, over cell means and not individual instances, and returns $\kappa_1 = 0.636 \cdot n^{0.81} \cdot d^{0.77} \cdot \alpha^{-0.15}$ at $R^2 = 0.90$. On the pivot instance it predicts $\kappa_1 = 11.28$ against the 13.0 empirical tuning gives, and neither value is feasible on the QPU at the κ_2 then in use. An early comparison of the two generators at $(d = 1.0, \alpha = 0.5)$ returned ratios of 0.89 and 1.02 at $n = 10$ and $n = 20$, and was read as clearing the generator of responsibility. It did not hold κ_2 constant across the two, and a controlled recalibration at $n = 10$, $\alpha = 0.259$, ten seeds

per cell, with κ_2 fixed for both, reverses that reading. The ratio $\kappa_1^{\text{Billionnet}}/\kappa_1^{\text{synthetic}}$ runs 0.787, 0.861, 0.805 and 0.919 at $d = 0.25, 0.50, 0.75$ and 1.00, farthest from unity at the density of every Billionnet test instance used here.

The synthetic generator assigns a linear profit to every item; the Billionnet protocol subjects the diagonal to d along with the pairwise terms, leaving a fraction $1 - d$ of items with no individual profit at all. Linear profits then have a coefficient of variation of 0.57 under the synthetic generator at every density, against 2.09 at $d = 0.25$ falling to 0.59 at $d = 1.00$ under the Billionnet protocol. The two distributions coincide at full density and diverge as density drops, and the calibrated ratio follows that shape.

The final regression, fitted on the Billionnet campaign over individual instances (1 437 of the 1 440, after excluding degenerate cells with $\kappa_1^* = 0$, all three at $n = 10$ and $d \leq 0.5$), yields

$$\kappa_1 = 0.395 \cdot n^{0.97} \cdot d^{1.04} \cdot \alpha^{-0.19}, \quad R^2 = 0.80, \quad \sigma_{\log} = 0.32. \quad (16)$$

Distributional regressors ($C/W_{\max}$, coefficients of variation of weights and profits, weight-profit correlation) raised R^2 from 0.80 to 0.81 and were dropped for parsimony.

The heuristic rescaling that Ohno et al. (2024) apply to their large instances, $\tau = (d/100)\sqrt{1/\alpha}$, is (15) with its exponents fixed a priori at 1 and $-1/2$ and its size term dropped. Dividing the calibrated κ_1 by τ does tighten their dispersion, $\sigma_{\log}$ falling from 0.724 to 0.505, without approaching invariance, and it fails on the tightness exponent, $-1/2$ against the -0.19 fitted here. Fitting the exponents instead of positing them takes the residual to 0.320.

3.4.3 Empirical anchoring: Method A

A residual standard deviation of $\sigma_{\log} = 0.32$ leaves the regression a factor $\exp(0.32) \approx 1.4$ of multiplicative uncertainty on κ_1^* , and the feasibility window on the pivot instance is narrower than that, a prediction short of the working value by less than a third returning no feasible run at all. Used as an absolute predictor, the formula is therefore of no use on hardware.

Eleven parameter pairs reach 99.9% on the pivot instance in exploratory runs, under chain strengths of 1.0, 2.0 and default torque compensation (UTC). The anchor retained is the one obtained under default torque compensation, the policy Section 3.3 fixes for every campaign that follows.

Anchoring on the empirically working pair $(\kappa_1, \kappa_2) = (13.0, 0.25)$ found on jeu_100_25_1 ($\alpha = 0.259$), the tightness exponent transports that anchor to the other instances of the same (n, d) through

$$\kappa_1(\alpha) = 13.0 \cdot (\alpha/0.259)^{c_\alpha}. \quad (17)$$

Two variants were run, one with $c_\alpha = -0.1924$ from the Billionnet regression (16) and one with $c_\alpha = -0.1451$ from the preliminary fit. Both retain the power-law form in α that Ohno et al. (2024) propose, with the exponent re-estimated here rather than taken from their Table VII, and

both leave κ_2 fixed at the anchor's value.

Section 4.4.5 reports what the correction achieves. It leaves untouched the difficulty that Section 3.4.4 takes up, a scalar correction along α moving κ_1 in one direction only, downward above the reference, while the instances that resist need it raised.

3.4.4 Two-dimensional grid search

Raising κ_2 from the anchor value to 0.4 across the ten-instance panel splits it. Two instances that no downward κ_1 correction had made feasible become feasible at once, while the loosest instance of the panel loses several points of objective and sees its chain-break fraction rise by an order of magnitude (Section 4.4.1). No single κ_2 serves a cell nominally homogeneous in (n, d) , and a second regression does not rescue the situation, returning robust directions on an R^2 near 0.25, κ_2 rising with size and density and falling with tightness. Directions established without a reliable level do not locate a search window, and the grid described below varies both coefficients instead.

Bayesian optimization would be the standard answer to a search this expensive over two parameters. It was not pursued, the grid having to produce a map of the viable region and not only its optimum.

The grid over κ_1 is multiplicative and recentred for every instance on the prediction of (16) for its own (n, d, α) . Nine values sit at offsets $\exp(s \sigma_{\log})$ for $s \in \{-2, -1.5, \dots, 2\}$ with $\sigma_{\log} = 0.32$, spanning a factor 1.9 either side of the prediction. Its width follows the dispersion of the predictor and not the spread of the instances, which an absolute grid cannot do, having no way to locate the centre for a given instance.

The grid over κ_2 is absolute instead, no rule being available to centre it. Seven values run from 0.1 to 0.6, the step tightening to 0.05 between 0.2 and 0.3 to resolve the neighbourhood of the anchor. The ceiling reflects what κ_2 does to the embedding. Expanding $\tilde{h}^2$ places a coupling $2\kappa_2 w_i w_j / W_{\max}^2$ on every pair of items, so κ_2 alone scales the clique that the chains have to carry, while κ_1 contributes linear biases and no coupling at all. Preliminary runs on the pivot instance at $\kappa_2 \geq 0.7$ returned infeasibility or chain-break fractions above the acceptance gate, and Chapter 4 shows the trend that produces it, the mean fraction rising by a factor near 1.7 for every 0.1 added beyond $\kappa_2 = 0.3$. One instance was carried past that ceiling nonetheless (Appendix A.5), and it did not behave as the pivot had.

The two axes give 63 cells per instance, each submitted ten times, for 630 QPU calls. Execution departed from that design as access ran short, the earliest block of 45 cells using five submissions and the coverage per instance ranging from 30 cells to 129; Chapter 4 and Appendix A.5 report what each instance actually received.

3.5 Evaluation metrics

The *feasibility rate* of a run is the fraction of its reads returning a sample that satisfies the capacity constraint. A cell (κ_1, κ_2) is called *feasible* when at least one read across its submissions is, a deliberately permissive test, an annealer being a sampler from which one keeps the best draw. The permissive definition excludes the regime where no setting of the penalty returns a usable solution at all, and Chapter 4 shows that regime to be large.

Pct_Opt is the ratio of the best objective observed to the proven optimum, in percent. Above 100% the objective has overcome a penalty too weak to hold, so the cell counts as a feasibility failure.

The *chain-break fraction* (cbf) is reported on every cell as the mean over its submissions, and cells above 0.1 are rejected. No universal threshold exists in the literature (Section 2.3.5); 0.1 is the working gate adopted here, and Section 4.4.1 shows that it decides the sign of one of the conclusions rather than merely tidying the data. A cell clearing both feasibility and this gate is *viable* throughout the remainder of this thesis.

Two statistics of the submissions coexist. Pct_Opt is the best of them, cbf their mean, so a cell is judged on its average chain reliability while its performance figure is the best of ten reprogrammings. Maximum chain length and maximum coupler amplitude after normalization accompany each campaign as secondary diagnostics.

3.6 Falsification protocols

3.6.1 P1 — The scaling law of Ohno et al.

The protocol tests $H_0 : c_n = 0.09$, the size exponent of Table VII in Ohno et al. (2024), against the regression (16) fitted here on an independent calibration campaign. The null is the exponent a practitioner applies who takes that table as a rule for setting a penalty, which is the use the rule is proposed for and the only one available on a pipeline without repair post-processing. The estimand tested here is accordingly the enforcement floor, the smallest κ_1 making the proven optimum the ground state of the QUBO, which is the quantity that decides whether a QPU returns a feasible sample at all. An order of magnitude separates the two published positions, the linear-in- n heuristic of Parizy and Togawa (2021) and Ohno's near-zero exponent being unable both to describe the same quantity.

The primary evidence is the Billionnet-protocol campaign of Section 3.1.2, 1 440 instances at $n \in \{10, 20\}$, with the preliminary study at $n \in \{10, 15, 18, 20\}$ as a coarser check across four sizes. Two bounds attach to that evidence, both of them consequences of the design rather than concessions made after the fact. Two points in n estimate a slope and not the shape of a curve, and the four-point check spans a single tightness cell at two densities, enough to corroborate a direction and not to refine an exponent.

The measurement is classical throughout. No embedding, chain, or coupler enters it, which

places any disagreement with Ohno in the QUBO formulation rather than in the annealer, their own evaluation being equally free of those constraints. The exponent governs the hardware side only afterwards, the κ_1 values sampled on the QPU being those (16) predicts for each instance, not points on a fixed grid.

Hardware supplies a second reading, coarser and of a different kind. The parameter pair that works at $n = 100$, $\kappa_1 \approx 13$ with $\kappa_2 \approx 0.25$, was also evaluated at $n = 10$ and on five instances at $n = 20$, on the same topology and with chain-break fractions below 0.05 throughout. No exponent is fitted from three sizes on one topology, and the campaign asks only whether a candidate value accommodates the range over which that setting stays usable.

The two rules parameterize different penalties, Ohno's λ weighting a squared deviation and κ_1 a linear one, which makes their prefactors incomparable. Their exponents are not, an exponent carrying no units, and Section 5.1 establishes the marginal-balance argument for both functional forms.

3.6.2 P2 — *The slack wall on Zephyr*

The topological obstruction is deterministic and can be established without QPU execution. At $n = 100$ the slack formulation (9) occupies $n + \lceil \log_2(C + 1) \rceil$ logical variables, which amounts to 110 on the pivot instance and to between 107 and 112 across the forty benchmark instances at that size. On the Zephyr working graph used here, the deterministic clique construction `busclique` returns a complete embedding up to K_{106} , at a maximum chain length of 13, and returns none at 107. Since the lower end of the benchmark range already exceeds that ceiling, clique-aware embedding is unavailable for every instance of the family at this size. Heuristic embedders are not bound by the uniform-chain construction and may still return a mapping by routing through irregular paths, which Section 4.3.2 evaluates against the chain lengths it produces.

The test runs both routes and records, for each, whether an embedding is returned, the chain lengths it produces, and whether any penalty setting yields a run at once feasible and inside the chain-break gate. The slack-free formulation (11) runs on the same instance at 100 variables for comparison, and the whole protocol repeats on Pegasus, whose K_{177} clique accommodates 110 with margin and therefore isolates whatever obstruction is not topological in origin.

3.6.3 P3 — *Parameter mapping and the transfer of calibrated parameters*

The grid of Section 3.4.4 yields, for each instance it covers, a map of `Pct_Opt` and `cbf` over (κ_1, κ_2) , and two hypotheses are read off it.

The first is that of Montañez-Barrera et al. (2024), that parameters tuned on one instance carry over to others of the same class. The test asks whether the pair optimal on the pivot lies inside the viable region of every other instance mapped.

The second is the assumption the calibration literature never states, that a penalty optimal

against a classical criterion is optimal on hardware. Settling it requires calibrating the same instance against both judges, which the exact oracle allows only as far as enumeration reaches. The comparison is therefore run at $n = 10$, applying Nelder-Mead twice to one instance, once against the bitmask oracle and once against QPU samples. At $n = 100$ no such pairing exists and the test is directional, asking whether the κ_1 that (16) predicts falls inside the QPU-viable region, an offset that Section 4.4.3 reads against the regression's own residual scale.

3.7 Scope

QPU access ran out on 15 June 2026, and the boundary shows in the results rather than in the design. The grid of Section 3.4.4 covers six of the ten `jeu_100_25_*` instances, all of them at $d = 25\%$, and the Pegasus campaign of Section 3.3 samples its cells once instead of ten times, leaving its figures to describe the shape of a failure without bounding it. Transfer to other densities was tested before the cutoff and is reported with the other outcomes in Chapter 4. Section 5.6 sets out what a renewed allocation should address first, and Appendix A.8 lists the archived instances, records, and code.

4 Results

4.1 Overview and reading guide

This chapter reports the three falsification protocols of Section 3.6, each stated as test, outcome, and immediate implication, with the interpretation against the gaps of Chapter 2 held over to Chapter 5.

4.2 P1 — The scaling exponent on instance size

The calibration campaigns rest on the bitmask oracle at $n \leq 20$ with no hardware involved, while the evaluation campaigns run on the QPU. The size exponent is measured on both sides, precisely on 1 437 classical calibrations and coarsely on hardware at three instance sizes.

The scaling law of Ohno et al. (2024), fitted on simulated annealing over the same Billionnet & Soutif benchmark family, predicts $c_n = 0.09$, so that the optimal penalty grows almost imperceptibly with instance size. The log-linear regression (15) of Section 3.4.2, fitted on the 1440-instance Billionnet campaign at $n \in \{10, 20\}$ (1437 points after excluding degenerate cells), yields the coefficients of Table A.3.

Testing the null hypothesis $H_0 : c_n = 0.09$ directly against this estimate gives

$$t = \frac{0.9723 - 0.09}{0.0342} \approx 25.8, \quad (18)$$

and the t of +28.43 reported in Table A.3 is the same statistic against a null of zero. The two agree, an exponent an order of magnitude above the prediction leaving no room for the choice of

null to matter. Standard errors are clustered by design cell, as the ten random seeds within each cell share identical (n, d, α) parameters and provide no independent information regarding the scaling exponents. A block bootstrap across the 144 cells corroborates this estimate to within 2%. Using the OLS standard error of 0.0244 yields a t -statistic of 36.1, and the 95% confidence interval for c_n , $[0.906, 1.036]$, encompasses unity.

Residuals are centred on zero with standard deviation 0.3203 and a largest absolute value of 4.39, so a few cells depart sharply from the fitted surface while the bulk stays within a factor $\exp(0.3203) \approx 1.38$ of the prediction. That figure recurs twice below, Section 3.4.4 converting it into the width of the QPU search window and Section 3.4.3 invoking it to rule the regression out as a point predictor, its multiplicative uncertainty exceeding the feasibility window observed on the pivot instance. Appendix A.7 plots the fit and the residual distribution.

The other two exponents place the disagreement precisely. The tightness exponent $c_\alpha = -0.1924$ sits within one standard error of Ohno’s -0.21 , at $t = 0.77$, so the two campaigns measure the same tightness dependence. The density exponent $c_d = 1.0416$ exceeds Ohno’s 0.84 at $t = 7.75$, a real but modest gap. Neither approaches the order of magnitude separating the size exponents, and the disagreement is concentrated on the one exponent Ohno’s rule places near zero. Table 1 situates the estimate against the two competing predictions identified in Section 2.5.2.

Table 1 – Scaling exponents on instance size: predictions and estimate.

Source	c_n	Basis
Ohno et al. (2024)	0.09	Fitted on simulated annealing
Parizy and Togawa (2021)	1 (implicit)	Posited as a starting heuristic
This thesis	0.97 (s.e. 0.034)	Fitted on a classical exact oracle

The estimate obtained here sits close to the Parizy heuristic and far from Ohno’s figure. This proximity should not be read as a confirmation of Parizy’s rule, since their $c_n = 1$ was never itself a fitted regression but a starting heuristic corrected per instance by a 50% incremental policy when replicas failed to converge (Section 2.5.2). The two published positions on this exponent are irreconcilable, and the present measurement decides in favour of near-linear growth.

The provenance of this estimate fixes what it does and does not contradict. It is fitted entirely on classical exact-oracle calibrations, where the criterion is whether the proven optimum is the ground state of the QUBO, and no embedding, chain, or coupler precision enters it. The near-linear growth of κ_1 with n is therefore a property of the QUBO formulation and not an artefact of the annealer. It is also a property of the enforcement floor, whereas the figure of Ohno et al. (2024) describes a performance optimum measured downstream of a repair procedure, and their own separation of $\lambda_{\text{critical}}$ from λ_{optimal} (Section 2.5.1) predicts that the two need not scale alike. The disagreement is accordingly not that their regression is wrong within its own setting. It is that the setting supplies a classical layer the QPU pipeline used here does without, and that a rule fitted with that layer in place understates by an order of magnitude the penalty a processor

needs when the QUBO reaches it intact.

The same parameter setting was evaluated at all three sizes on this topology, $\kappa_1 \approx 13$ with $\kappa_2 \approx 0.25$, and returns 28.0% of the proven optimum at $n = 10$, 17.2% on average over five instances at $n = 20$, and 99.9% at $n = 100$, with chain-break fractions below 0.05 throughout. Chain integrity holding in all three, the separation is attributable to the calibration alone. The settings that do work at the smaller sizes lie far below, $\kappa_1 \leq 2.8$ reaching the optimum at $n = 10$, and $\kappa_1 = 5.0$ reaching 94 to 100% at $n = 20$, a size at which $\kappa_1 = 1.05$ returns nothing feasible. Since κ_2 was not held constant across those campaigns, no exponent follows from them.

One instance at the smallest size, and a change of generator between $n = 20$ and $n = 100$, put an exponent estimate out of reach here. The three sizes exclude $c_n = 0.09$, which would place the $n = 10$ optimum near 11, the value at which the QPU misses the optimum by 72 percentage points with its chains intact.

Three bounds on the classical estimate close the protocol. It rests on two sizes, so it fits a slope and does not establish the shape of the curve, and the cross-check at $n \in \{10, 15, 18, 20\}$ spans two densities and two tightness values, enough to confirm a direction and not to refine an exponent. It is also fitted at a fixed κ_2 , which forces κ_1 to carry any size dependence the pair holds jointly, so the exponent estimated is that of κ_1 under that constraint. The two campaigns bound the effect: the synthetic campaign at $\kappa_2 = 0.1$ returns $c_n = 0.81$ and the Billionnet campaign at $\kappa_2 = 0.25$ returns 0.972, a displacement of 0.16 across a factor 2.5 in κ_2 and a change of generator, against the order of magnitude separating either figure from 0.09. None of the three displaces the conclusion, and all three confine the claim to what the design supports.

4.3 P2 — The slack wall: two levels of obstruction

Section 3.6.2 predicted failure for the slack formulation at $n = 100$ on Zephyr, where the pivot instance needs 110 logical variables and no instance of the benchmark at that size needs fewer than 107. Clique construction on the working graph used here returns a complete embedding up to K_{106} , at a maximum chain length of 13, and none at 107. An exploratory campaign of 575 QPU runs on jeu_100_25_1, crossing both topologies with both formulations, tests the prediction and locates the obstruction more precisely than the topological argument allows.

4.3.1 The four configurations

Table 2 – Viable runs by topology and formulation, instance `jeu_100_25_1` ($n = 100$). A run is viable if it returns a feasible solution with $\text{CBF} \leq 0.1$.

Topology	Formulation	Variables	Runs	Viable	Best viable Pct_Opt
Zephyr	slack	110	24	0	—
Zephyr	unbalanced	100	260	140	99.9%
Pegasus	slack	110	287	0	—
Pegasus	unbalanced	100	4	4	85.2%

The outcome separates by formulation and not by topology (Table 2). Across 311 runs spanning two embedders, three chain-strength policies, and penalty coefficients over two orders of magnitude, the slack formulation returned no viable run on either platform, while the unbalanced formulation returned them on both.

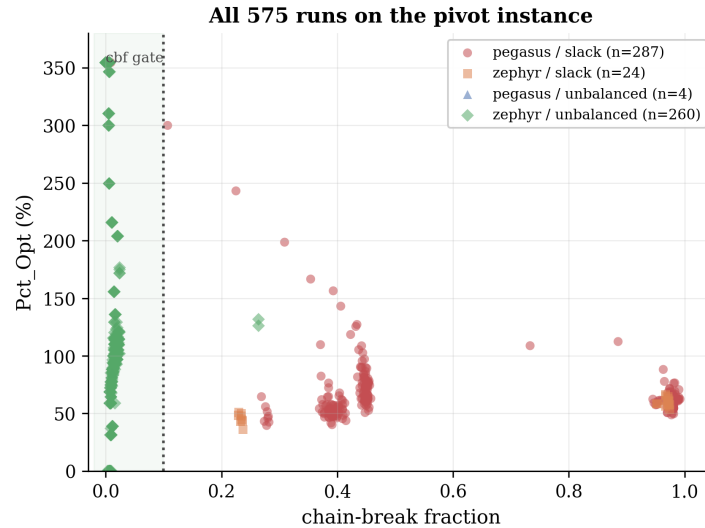


Figure 1 – All 575 exploratory runs on `jeu_100_25_1`, positioned by chain-break fraction and Pct_Opt. The acceptance zone, at $\text{cbf} \leq 0.1$, holds no slack run on either topology.

Figure 1 positions all 575 runs by chain-break fraction and objective. The slack runs separate into two clouds, one near $\text{cbf} = 0.45$ and one near 0.97, which Section 4.3.3 traces to the two chain-strength policies; neither reaches the acceptance zone.

The four Pegasus runs under the unbalanced formulation are a positive control rather than a calibration sweep, and they sample the one parameter point, $(\kappa_1, \kappa_2) = (13.0, 0.2)$, that the Zephyr campaigns also cover on this instance. There, Zephyr reaches 99.9% of the proven optimum at a chain-break fraction of 0.014 over ten submissions, against 85.2% over four on Pegasus. Four runs at one point support no verdict on the two generations, and what they do establish is that the slack-free formulation is viable on both.

4.3.2 Refining the topological prediction

The Zephyr/slack cell holds executed runs where the topological argument predicted none, and the embedder record explains them. All 24 come from the heuristic embedder `minorminer`, while `busclique`, present in every other Zephyr campaign of this dataset, is absent from this one. Clique-aware embedding evidently failed at 110 variables, as the ceiling predicts, and the heuristic route took over. Free of the uniform-chain construction, heuristic embedders route through irregular paths and succeed where clique construction fails.

What that freedom costs is measurable on Pegasus, where both embedders return a mapping for the same 110-variable slack problem. `busclique` holds the longest chain to 11 qubits, `minorminer` stretches it to 23 (Table 9). On Zephyr the heuristic route reaches 20, against 12 for the clique embedding of the 100-variable unbalanced model, and chain fidelity falls as chain length rises (Section 2.3.5).

Chain integrity does not withstand those lengths. Across the 24 Zephyr/slack runs the median chain-break fraction is 0.967, so roughly 106 of the 110 chains fail in a typical read, and the best read of the entire campaign still leaves 25 of them broken at a fraction of 0.229, twice the acceptance gate.

Protocol P2 therefore reaches its conclusion by a route the prediction stated too sharply. The slack formulation at $n = 100$ is not unembeddable on Zephyr; it is beyond clique-aware embedding, and the heuristic alternative produces chain lengths at which none of the three strength policies tested keeps the breakage below the gate. The obstruction admits degrees.

4.3.3 The electronic wall on Pegasus

Pegasus isolates the second level. Its K_{177} clique takes the 110 variables with margin, and no topological argument applies. None of the 287 runs is viable.

Table 3 – The chain-strength dilemma, slack formulation on Pegasus.

Chain-strength policy	Runs	Feasible	Median CBF
Fixed ($J_F \in \{1.0, 2.0\}$)	128	28	0.98
<code>uniform_torque_compensation</code>	159	1	0.41

The failure takes two forms according to the chain-strength policy (Table 3). A fixed strength returns feasible samples on 28 runs of 128, at a median chain-break fraction of 0.98, which on 110 logical variables means roughly 108 chains failing in a typical read; the configurations reported at that level are not assignments of the logical variables in any meaningful sense. Torque compensation more than halves the breakage and takes feasibility with it, one run out of 159.

Figure 2 traces the mechanism over 65 values of λ between 0 and 50. At $\lambda = 0$ the chains hold, no penalty competing with the ferromagnetic coupling, and the annealer returns all one

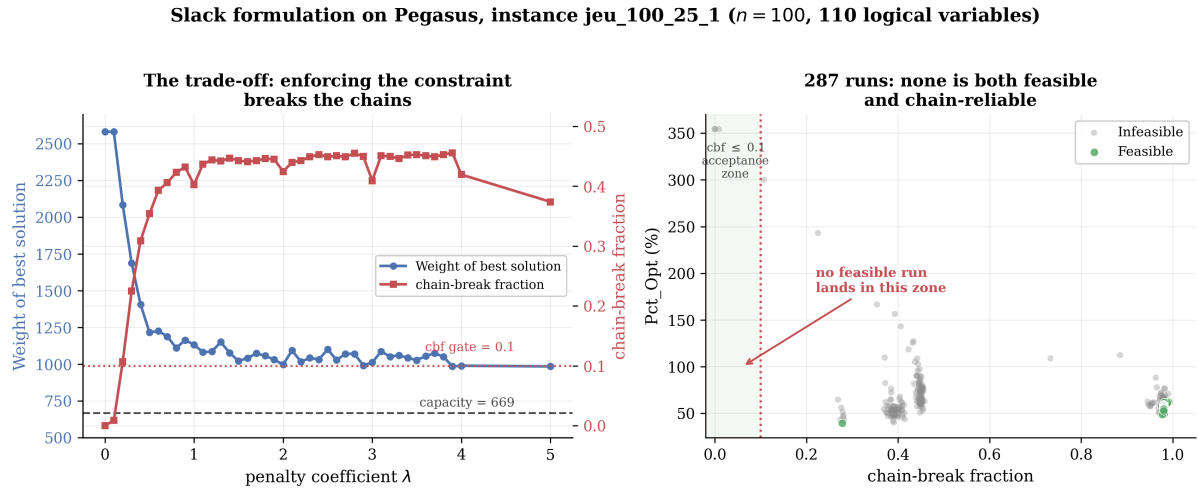


Figure 2 – The trade-off between constraint enforcement and chain integrity, slack formulation on Pegasus, instance jeu_100_25_1. Left: weight of the best returned solution and chain-break fraction as functions of λ . Right: all 287 runs positioned by CBF and Pct_Opt; the acceptance zone contains no feasible run. The complete sweep appears in Appendix A.2

hundred items for a weight of 2 582, which is the sum of the weights and the expected answer to an unconstrained maximization over non-negative profits. Raising λ drives the weight down monotonically and the chain-break fraction up, until both stall, the weight between 980 and 1 050 and the fraction near 0.45. The capacity of 669 is not reached at any penalty in the sweep. The chain-break fraction therefore crosses the acceptance gate while the solution weight remains half again above the bound, and no larger penalty closes the difference.

This is the dynamic-range mechanism of **rosenberg2016** (Section ??), observed directly on the QKP. Objective coefficients, penalty coefficients, and chain strengths draw on one precision budget, and the coefficient magnitudes of the slack encoding, 10^5 to 10^6 against an objective bounded by 100 (Section 2.4.3), exhaust it.²

4.4 P3 — Parameter transfer on physical hardware

Protocol P3 tests the two hypotheses of Section 3.6.3, that parameters tuned on one instance generalize to others of the same class (Montañez-Barrera et al., 2024), and that parameters optimal against a classical exact oracle transfer to the QPU.

The two-dimensional grid covers 378 cells over six of the ten jeu_100_25_* instances, at ten submissions per cell except for the earliest block of 45 cells, which used five, for 3 555 QPU calls in all. The Method A campaigns of Section 3.4.3 cover all ten instances at three configurations and four values of κ_2 . A transfer campaign carries parameters calibrated at $n = 10$ onto five instances at $n = 20$. A separate campaign carries the regression prediction onto densities other than 25%.

2. Each cell of the Pegasus campaigns was sampled once, against ten times elsewhere, so the figures characterize the shape of a failure and bound no individual cell. The penalty swept is the physical λ of (?), which carries no reading in the κ of the unbalanced model.

4.4.1 The chain-break gate as a discriminator

The grid supports opposite conclusions depending on whether the chain-break gate is applied.

Over feasible cells alone, Pct_Opt rises monotonically with κ_2 , from 42.7% at $\kappa_2 = 0.10$ to 81.0% at $\kappa_2 = 0.60$. Without further filtering, one would naturally conclude that the $\kappa_2 = 0.25$ retained in calibration is too low at $n = 100$, implying a scale dependence in κ_2 that requires its own transfer rule.

The chain-break fraction over those same cells contradicts this reading. It rises from 0.008 to 0.253, crossing the acceptance gate between $\kappa_2 = 0.40$ and 0.50; by $\kappa_2 = 0.60$, only 8 of the 36 sampled cells remain below the threshold. Applying the gate reverses the trend. Among viable cells, both feasible and chain-reliable, Pct_Opt peaks at $\kappa_2 = 0.30$ (60.4%) before falling off, and the number of viable cells collapses from 27/27 at $\kappa_2 = 0.10$ to 6/25 at $\kappa_2 = 0.60$. Cell means of κ_1 range from 11.2 to 14.0 across the seven levels, confirming that the linear coefficient does not confound this comparison.

Table 4 – Mean Pct_Opt by κ_2 , with and without the chain-break acceptance gate. Grid campaigns at $n = 100$, six instances, 378 cells.

κ_2	0.10	0.20	0.25	0.30	0.40	0.50	0.60
Mean CBF	0.008	0.009	0.015	0.054	0.090	0.172	0.253
Pct_Opt, feasible cells only	42.7	49.5	52.3	69.7	69.0	72.7	81.0
Viable cells (of feasible)	27/27	35/35	44/46	13/18	29/44	17/38	6/25
Pct_Opt, viable cells only	42.7	49.5	50.5	60.4	56.5	50.2	51.5

The averages of Table 4 conceal a split. Raising κ_2 from the anchor value to 0.40 across the ten-instance panel turns jeu_100_25_6 and jeu_100_25_8 feasible for the first time, and lifts jeu_100_25_3 from 32.8% to 96.9%, while jeu_100_25_5, the loosest of the panel at $\alpha = 0.934$, falls from 99.2% to 95.0% and sees its chain-break fraction rise from 0.007 to 0.132.

The apparent gain at high κ_2 comes from samples whose chains have broken, which carry no well-defined assignment of the logical variables. This mechanism echoes what Section 4.3.3 established for the slack formulation, here along the κ_2 axis and not λ , and its recurrence marks the dynamic-range constraint (Section 2.3.5) as one that governs the whole calibration.

Section 3.4.4 fixed the κ_2 ceiling at 0.6 from preliminary runs on the pivot instance. Three cells at $\kappa_2 = 0.75$ on jeu_100_25_3 were sampled beyond it, two of them viable at 90.2% and 91.5% with chain-break fractions of 0.067 and 0.018, neither improving on that instance's best cell at $\kappa_2 = 0.30$. The ceiling therefore does not hold on _3, which carries the lowest tightness of the panel at $\alpha = 0.063$. That is the direction the auxiliary regression on κ_2 predicts, its negative exponent on α placing the largest values at the tight end, and one instance is not enough to establish where the ceiling sits in general.

4.4.2 Transfer across instances

The first hypothesis holds that parameters tuned on one instance carry over to others of the same class. Testing it directly requires one pair and several instances, which Table 5 reports. The pair is the one Nelder-Mead selects against QPU samples on the pivot, $(\kappa_1, \kappa_2) = (13.44, 0.285)$, reaching 99.9% of the proven optimum there over 22 hardware evaluations.

Table 5 – The pivot’s calibrated pair $(\kappa_1, \kappa_2) = (13.44, 0.285)$ applied unchanged to six jeu_100_25_* instances, ordered by tightness. Weight is that of the best returned solution.

Instance	α	Pct_Opt	Weight / capacity	Feasible	CBF
_3	0.063	9.8%	13 / 156	yes	0.000
_1	0.259	99.9%	668 / 669	yes	0.010
_8	0.262	107.5%	671 / 619	no	0.020
_6	0.540	110.8%	1362 / 1236	no	0.000
_2	0.855	94.3%	2227 / 2366	yes	0.020
_5	0.934	92.9%	2327 / 2536	yes	0.010

On _8 and _6 the penalty is too weak and the objective crosses it, the returned weight exceeding capacity by 8% and 10%. On _3 it is too strong, the sampler returning a knapsack filled to 13 of its 156 units of capacity for 9.8% of the optimum. Three instances lie between, at 92.9% to 99.9%. Chain-break fractions stay at or below 0.020 throughout, so the calibration alone separates the six outcomes. Lowering the pair to $(13.0, 0.2)$ leaves the four working instances unchanged and worsens the two that overshoot, to 114.5% and 115.8%.

Both failures follow from the calibrated values the grid returns for each instance separately, which Table 6 reports.

Table 6 – Best viable configuration per instance, grid campaigns at $n = 100$, $d = 25\%$. Where several cells reach the same objective, the entry reported is the one with the lowest chain-break fraction among them, so the κ_1 listed marks a point inside the viable band rather than a unique optimum.

Instance	κ_1	κ_2	Pct_Opt	CBF
jeu_100_25_1	14.53	0.20	99.9%	0.008
jeu_100_25_2	9.84	0.10	100.0%	0.007
jeu_100_25_3	11.81	0.30	99.1%	0.006
jeu_100_25_4	11.81	0.10	100.0%	0.020
jeu_100_25_6	14.81	0.20	100.0%	0.010
jeu_100_25_8	14.50	0.25	100.0%	0.010

_3 calibrates at $\kappa_1 = 11.81$ and receives 13.44; _6 calibrates at 14.81 and receives the same. A single pair cannot serve a class whose members require values on either side of it, and the hypothesis of Montañez-Barrera et al. (2024) fails in the form where one tuned pair would. Four of the six instances reach the proven optimum and the other two come within one percentage point, the optimal κ_1 spanning $[9.84, 14.81]$ and the optimal κ_2 spanning $[0.10, 0.30]$.

Converted through (13), the pair Montañez-Barrera et al. (2024) obtain on a thirteen-item problem reads (0.48, 0.93) against the $[9.84, 14.81] \times [0.10, 0.30]$ the six instances require here, so their κ_1 sits a factor twenty below the smallest value viable at $n = 100$ and their κ_2 a factor three above the largest. Neither coordinate transfers, and the ordering of the two coefficients reverses.

The weaker form of the hypothesis holds. The pivot's optimum falls inside the viable region of four of the other five instances, missing jeu_100_25_6 only because that instance's viable range begins at 14.81, and fixing $\kappa_2 = 0.25$ leaves 44 of the 76 cells sampled at that level viable, 44 of the 46 that were feasible. A region transfers rather than a point, and its width makes calibration tractable at all.

A correction along one coordinate does not reach two of the six, and the record locates the difficulty precisely. Across the anchored campaigns, which hold κ_1 within $[11.29, 13.00]$, jeu_100_25_6 and jeu_100_25_8 return an infeasible solution at every level of $\kappa_2 \in \{0.10, 0.20, 0.25\}$, with Pct_Opt between 114% and 172%, the signature of an objective the penalty no longer holds. On _8 the tightness correction is inert, its $\alpha = 0.262$ sitting within 0.003 of the anchor's own, so the three κ_1 values tested span 0.025 and behave identically at each level.

Two moves recover both instances, and they act on different axes. Raising κ_2 to 0.40 at the anchor's κ_1 turns _8 feasible at 98.9% over ten submissions of ten, and _6 at 99.3%. Holding κ_2 at 0.20 or 0.25 and raising κ_1 instead reaches the proven optimum on both, at 14.81 on _6 and 14.50 on _8 (Table 6). The anchor fails on these two because it lies below their viable range in κ_1 , and the tightness correction of (17) lowers it further, to 11.29 and 11.69 on _6, where at $\kappa_2 = 0.40$ only the unadjusted anchor returns a feasible sample.

Preliminary campaigns confined to $\kappa_1 \in [13, 14]$ around the formula's prediction returned viable results at 92–99% on four of six harder instances and no feasible run on _6 or _8. Both are recovered once the window opens to $\pm 2\sigma_{\log}$, at a multiplier of 1.616 on the upper side, while jeu_100_25_3 finds its best cell at $-0.5\sigma_{\log}$, so the three instances that need the window widened need it widened in opposite directions.

Across the ten instances of the family, taking the grid and the anchored campaigns together, every one admits at least one viable configuration (Table 7). Eight reach the proven optimum exactly, and the two that do not, jeu_100_25_1 and jeu_100_25_3, fall short by 0.1 and 0.9 percentage points.

4.4.3 *Transfer from classical calibration to hardware*

The second hypothesis asks whether a penalty calibrated against a classical exact oracle is the penalty the QPU needs.

Regression (16), fitted at $n \in \{10, 20\}$ with no hardware involved, predicts $\kappa_1 = 10.63$ for the pivot instance, where the grid finds its best viable cell at 14.53. The ratio, 1.367, sits within one

Table 7 – Best viable configuration found on each of the ten `jeu_100_25_*` instances, across the grid and the anchored campaigns.

Instance	α	κ_1	κ_2	Pct_Opt	Campaign
_3	0.063	11.81	0.30	99.1%	grid
_7	0.190	13.80	0.20	100.0%	anchored
_1	0.259	14.53	0.20	99.9%	grid
_8	0.262	14.50	0.25	100.0%	grid
_10	0.399	13.00	0.20	100.0%	anchored
_6	0.540	14.81	0.20	100.0%	grid
_9	0.568	11.18	0.25	100.0%	anchored
_4	0.762	11.81	0.10	100.0%	grid
_2	0.855	9.84	0.10	100.0%	grid
_5	0.934	10.16	0.10	100.0%	anchored

per cent of $\exp(\sigma_{\log}) = 1.378$, so the prediction misses by exactly the regression’s own residual scale. Across the six mapped instances the ratio runs from 0.846 on `jeu_100_25_3`, where the prediction overshoots, to 1.604 on `jeu_100_25_6`, with a geometric mean of 1.26. The offset is therefore not a uniform bias in one direction, and its spread is what fixes the window: all six fall inside $\pm 2\sigma_{\log}$, and `_6` already falls outside $\pm\sigma_{\log}$. The regression places the window; it cannot serve as a point predictor.

Two limits attach to the measurement. Grid cells advance in multiplicative steps of $\exp(0.5 \sigma_{\log}) \approx 1.17$, so the offset is quantized at that resolution, which is why three instances return the identical ratio 1.367, their best cells all falling on the $s = +1$ node. And the classical calibration is fitted at $n \in \{10, 20\}$ while the QPU evaluation runs at $n = 100$, so an offset measured here could equally reflect divergence between the two criteria or extrapolation across a factor-five range in size.

A paired calibration separates them in part. On one generated instance at $n = 10$, $d = 25\%$, $\alpha \approx 0.26$, the bitmask oracle returns $\kappa_1^* = 2.00$ while Nelder-Mead against QPU samples on the same instance reaches the proven optimum at $\kappa_1 = 1.0$ and again at 2.64, placing the classical value inside the hardware-viable interval. At a size where both criteria can be applied to the same problem, they agree. One instance does not settle the question, and it does shift the weight of the $n = 100$ offset onto extrapolation, away from a divergence between the classical and physical criteria. The third gap of Chapter 2 is narrowed, and the paired grid that would close it is out of reach at $n = 100$, where no exact oracle exists.

4.4.4 Transfer to other densities

The grid campaigns cover $d = 25\%$ alone. A separate campaign carried the prediction of (16) onto three instances at each of $d \in \{0.50, 0.75, 1.00\}$, alongside the unadjusted anchor, the tightness-corrected variants, and a fourth rule prescribing κ_1 six to ten times higher, over four values of κ_2 from 0.10 to 0.50, with a follow-up extending the sweep to 0.90 on the same

instances. No configuration returned a feasible solution at $d = 0.50$ or $d = 0.75$. The 54 cells of the first campaign at those two densities and the 54 of the second returned none, across κ_1 values from 11.4 to 28.6 and every κ_2 tested, while the mean chain-break fraction stayed between 0.003 and 0.014 over the whole range up to $\kappa_2 = 0.6$, an order of magnitude below the acceptance gate, so chain integrity is not the obstacle.

Full density fails differently. Twelve cells there return feasible solutions, all at κ_1 between 79.3 and 125.7, and eight of the twelve return the empty knapsack, a weight of zero against capacities of 574 and 100, three more returning one or two items for an objective of 44 against a proven optimum of 225 434. Only one cell, at $\kappa_1 = 79.3$ and $\kappa_2 = 0.50$, returns a partially filled solution, reaching 38.6% of the optimum on jeu_100_100_3. At these magnitudes the penalty cancels the objective outright, leaving the empty knapsack the cheapest configuration available, and chain-break fractions of 0.003 to 0.010 confirm that nothing physical prevents the sampler from returning it. Both regimes that Section 2.4.6 describes from the literature therefore appear at fixed size on hardware, one at each end of the density range, and no setting tested occupies the interval between them where the constraint binds and the objective survives.

4.4.5 The tightness correction in retrospect

Table 8 – Best Pct_Opt over viable cells, by instance and configuration, ordered by tightness. Each figure is the best the configuration reaches over the four values of κ_2 tested; the anchor applies $\kappa_1 = 13.0$ throughout. A dash marks an instance on which the configuration returned no viable cell.

Instance	α	κ_1 applied		Best Pct_Opt (%)		
		bill	synth	ref	corr_bill	corr_synth
_3	0.063	17.08	15.97	96.9	21.6	22.8
_7	0.190	13.80	13.60	100.0	100.0	100.0
_1	0.259	13.00	13.00	99.9	99.9	99.9
_8	0.262	12.98	12.98	98.9	98.9	98.9
_10	0.399	11.97	12.21	100.0	100.0	100.0
_6	0.540	11.29	11.69	99.3	—	—
_9	0.568	11.18	11.60	100.0	100.0	100.0
_4	0.762	10.56	11.12	99.6	100.0	100.0
_2	0.855	10.33	10.93	98.3	100.0	100.0
_5	0.934	10.16	10.79	94.8	100.0	100.0

The empirical anchoring of Section 3.4.3 moves κ_1 along α from a reference that works, raising it below $\alpha_{\text{ref}} = 0.259$ and lowering it above. Averaged over viable cells the unadjusted anchor leads at every level of κ_2 (Appendix A.6). Read instance by instance, as Table 8 reports, the correction divides the panel along α_{ref} .

Above α_{ref} the correction lowers κ_1 and gains 0.4 points on jeu_100_25_4, 1.7 on _2 and 5.2 on _5, the gain growing with tightness. Below it the correction raises κ_1 to 17.08 and jeu_100_25_3 falls from 96.9% to 21.6%. On jeu_100_25_6 it removes feasibility altogether,

the anchor reaching 99.3% at $\kappa_2 = 0.40$ while neither corrected variant returns a feasible sample. Five instances between the two extremes return the same figure whichever configuration is applied, and _6 sits among them in tightness while behaving like the tight end.

Table 6 accounts for the split. The calibrated κ_1 rises with α before falling, peaking between $\alpha = 0.26$ and 0.54 , whereas the correction declines monotonically by construction. On the descending branch the two agree and the correction earns its points. On the ascending branch it pushes κ_1 to 17.08 while the grid locates the optimum at 11.81, a third too high and in the wrong direction. The exponent $c_\alpha = -0.19$ therefore describes half the domain.

4.5 Hardware characterization

The unbalanced formulation at 100 logical variables occupies 1 149 physical qubits on Zephyr, roughly a quarter of the working graph, at a mean chain length of 11.5. The slack formulation at 110 variables occupies 1 617 under the heuristic route, so ten per cent more logical variables cost forty-one per cent more hardware.

Table 9 – Chain profiles by topology, formulation, and embedder, instance jeu_100_25_1.

Topology	Formulation	Embedder	Variables	Max chain	Mean chain
Zephyr	unbalanced	busclique	100	12	11.5
Zephyr	slack	minorminer	110	20	14.7
Pegasus	slack	busclique	110	11	10.8
Pegasus	slack	minorminer	110	23	14.7

Only the Pegasus pair isolates the embedder, holding topology and problem fixed. There busclique halves the longest chain and saves 27% of the physical qubits, which is the basis for the compilation policy of Section 3.3. The Zephyr rows differ in formulation as well as in embedder and cannot be read the same way.

4.6 Chapter summary

All ten jeu_100_25_* instances admit a penalty setting under which the QPU returns a feasible solution with reliable chains, eight at the proven optimum and two within one percentage point. P1 rejects $c_n = 0.09$ at $t = 25.8$ and places the size exponent at 0.97 (s.e. 0.034). The measurement being classical throughout, it describes the QUBO formulation and says nothing about the annealer. P2 finds no viable run of the slack formulation in 311 attempts, obstructed by the clique ceiling on Zephyr and by the precision budget on Pegasus. P3 finds no single parameter pair serving the whole class and a shared viable region that does, the classical regression placing a window of $\pm 2\sigma_{\log}$ without serving as a point predictor. Cross-cutting all three, the chain-break gate decides the sign of the conclusion on κ_2 .

5 Discussion

Chapter 2 identified three gaps in the calibration literature for the QKP on annealing hardware. Chapter 4 supplied an independent re-estimate of the size exponent, characterized the slack encoding on both topologies, and returned a directional answer on the third. This chapter asks what those outcomes share, how far the calibration architecture generalizes beyond the QKP, and what the empirical work does not support.

5.1 The nature of the scaling exponent

The exponent $c_n = 0.97$ of Section 4.2 was fitted against the classical exact oracle, whose criterion is whether the proven optimum is the ground state of the QUBO. No chain, no embedding and no coupler precision entered that measurement, so an account of the disagreement with Ohno that appeals to chain length or to the shared precision budget of Section 2.3.5 cannot be the explanation. The explanation has to be visible in the formulation.

It is, and it can be stated quantitatively. For the proven optimum to be the ground state of the QUBO, the penalty has to outweigh the objective gain of every infeasible configuration, including those that draw on the whole item set. Adding an item to a selected set gains, in the normalized objective, on the order of $n \cdot d \cdot \mathbb{E}[p_{ij}]/P_{\max}$, since each of the items available contributes a pairwise term with probability d ; the same addition costs $\kappa_1 \cdot \mathbb{E}[w]/W_{\max}$ in penalty. Balancing the two,

$$\kappa_1 \sim n d \frac{\mathbb{E}[p_{ij}] W_{\max}}{P_{\max} \mathbb{E}[w]}, \quad (19)$$

which predicts $c_n = 1$ and $c_d = 1$ against the 0.972 and 1.042 measured, the first within the 95% interval the calibration campaign returns. Under the generation protocol of Section 2.1.3, $\mathbb{E}[p_{ij}]/P_{\max} \approx 0.5$ and $W_{\max}/\mathbb{E}[w] \approx 2.0$, placing the prefactor (19) implies near 1.0, which overshoots the fitted $A = 0.395$ by a factor near two and a half, as an order-of-magnitude balance that ignores the distribution of marginal gains should.

The same balance applies to the squared penalty, which is what makes the two exponents comparable. Under (9) the slack bits cover $\{0, \dots, C\}$ exactly, so the deviation vanishes on the feasible side and a configuration exceeding capacity by v costs λv^2 . Adding k items to a set at the capacity boundary gives $v \approx k \mathbb{E}[w]$ and gains $k n d \mathbb{E}[p_{ij}]$, the same marginal gain that enters (19), so the requirement reads $\lambda \gtrsim n d \mathbb{E}[p_{ij}]/(k \mathbb{E}[w]^2)$ and binds at $k = 1$, returning $c_n = 1$ and $c_d = 1$ for the squared form as well. Counting the $\binom{k}{2}$ pairs the added items form among themselves replaces n/k by $n/k + 1/2$ and leaves the binding value of k unchanged. Whichever form the penalty takes, the coefficient has to dominate a marginal gain that grows with the number of items available, and the two rules therefore make commensurable claims about how that coefficient scales.

Removing the regularization bias of Section 3.4.1 moves the two exponents in opposite directions relative to that prediction, c_n to 1.004 and c_d to 1.097. The balance therefore accounts

for the size dependence, which no version of the fit distinguishes from unity, and less closely for density, where a factor it omits is presumably at work.

Substituting the number of items in the proven optimum for n in this argument does not survive testing. Recomputing $|S|$ on all 1 437 calibrated instances and refitting $\log \kappa_1^*$ on $\log |S|$ and $\log d$ returns $R^2 = 0.60$ against the 0.80 of (16), and entering both n and $|S|$ gives $|S|$ a negative coefficient. The reason is visible in the data: $|S|/n$ runs from 0.31 at low tightness to 0.92 at high, so a penalty proportional to $|S|$ would rise with α , while the calibrated one falls. The penalty has to dominate the worst violation available, sized by n , and not the optimum, sized by $|S|$.

Parizy and Togawa (2021) state the same reasoning as their motivation for a linear-in- n penalty on a different Ising machine (Section 2.5.2). Bontekoe et al. (2023) supply an independent measurement of it. Their Table 6 reports the empirically optimal penalty for each cell of the Billionnet benchmark under simulated annealing, and at $d = 50\%$ those values run 4, 8 and 15 at $n = 100, 200$ and 300 , a log-log slope near 1.2; at $d = 25\%$ the slope over the same range is near 1.1, and across densities at $n = 100$ the exponent is near 0.87. Their formulation is slack-type and their sampler is classical, so the magnitudes do not transfer, but an exponent carries no units and theirs falls where this thesis places its own.

Two questions separate at this floor. The first asks whether the annealer requires more than the combinatorial minimum, a hardware surcharge above what the formulation alone dictates, and Section 4.4.3 answers it only directionally. The QPU optimum on the pivot exceeds the classical prediction by a factor 1.367, the ratio running from 0.846 to 1.604 across the six mapped instances, which is compatible with such a surcharge and equally with extrapolation error in a regression fitted at $n \in \{10, 20\}$ and applied at $n = 100$. The second asks whether a solver may operate below the floor, and Ohno's rule answers it affirmatively for a pipeline that repairs infeasible samples classically.

Jeong et al. (2025) report a disagreement of similar shape elsewhere in the annealing literature. Fitting the chain strength required to preserve chain integrity against chain length, they obtain an exponent between 0.82 and 0.85 against the 0.5 a simple theoretical argument gives. Chain strength and penalty strength are different quantities measured under different conditions, so nothing follows about a shared cause, though in both cases an analytical prediction understates what a physically realized system turns out to require.

The position of the calibrated pair on its own parabola explains the role of the second coefficient. The vertex of (11) sits at $\tilde{h}^* = \kappa_1/(2\kappa_2)$, which the pivot's calibrated pair puts at 26 in units of $W_{\max}$, against a largest attainable slack of $C/W_{\max} = 13.38$. The vertex lies outside the reachable range, so over every configuration the sampler can return the penalty decreases monotonically in the slack, and near the capacity boundary, the point at which the constraint is decided, the quadratic term contributes under a tenth of the linear one. Enforcement is carried by κ_1 almost alone.

κ_2 carries load on the embedding instead. Expanding $\tilde{h}^2$ places a coupling $2\kappa_2 w_i w_j / W_{\max}^2$

on every pair of items, bounded by $2\kappa_2$, so the penalty clique reaches 0.50 at the calibrated $\kappa_2 = 0.25$ and 1.20 at the ceiling of 0.6, against objective couplers normalized into $[-1, 1]$. The ceiling Section 3.4.4 fixes empirically is where the penalty's own clique overtakes the objective it is supposed to constrain, and the chain-break trend of Section 4.4.1 follows the same quantity. This also accounts for the failure of transfer measured in Section 4.4.2: pair tuned at $n = 13$, with a negligible clique, is under no pressure to keep κ_2 small, which is why it arrives at $n = 100$ with the two coefficients in the opposite order.

5.2 The instance space is not uniform

Calibration difficulty is distributed unevenly across the ten instances of the family, and the pattern runs against the functional form of (16).

`jeu_100_25_3`, at $\alpha = 0.063$, received 144 configurations across the grid and the anchored campaigns, more than any other instance, and returned 119 viable cells and none at the proven optimum, its best result standing at 99.1%. Every other instance of the family reaches the optimum exactly. Chain integrity almost never stops it, 142 of those 144 configurations clearing the chain-break gate. Its resistance comes from the placement of the penalty.

The calibrated κ_1 of Table 6 rises with α , from 11.81 at $\alpha = 0.063$ to 14.81 at $\alpha = 0.540$, and falls thereafter to 9.84 at $\alpha = 0.855$. A multiplicative form with a single exponent cannot reproduce a maximum, and $c_\alpha = -0.19$ imposes a monotone decline. Section 4.4.5 measured what that costs: the tightness correction lowers κ_1 on the descending branch and gains points, and raises it to 17.08 on the ascending branch, costing `jeu_100_25_3` seventy-five points while the grid locates its optimum at 11.81.

The same asymmetry accounts for the two instances Section 4.4.2 reports as unreachable by the anchored campaigns. `jeu_100_25_8` at $\alpha = 0.262$ and `jeu_100_25_6` at $\alpha = 0.540$ bracket the maximum of the calibrated κ_1 , so both sit on the ascending branch, along which the correction pulls in the wrong direction, and both are recovered by the grid at κ_1 above the anchor rather than below it. Section 4.4.2 records this as a failure of a one-dimensional rule; it is a failure of the functional form.

If the tightness dependence were a proxy for the size of the optimal solution, replacing α by the number of selected items would improve the fit; Section 5.1 reports that it degrades it, and that $|S|$ carries a negative coefficient once n is present. The non-monotone behaviour is therefore not an artefact of using α in place of $|S|$. What remains is either a genuinely non-monotone dependence on tightness, which no multiplicative form in (n, d, α) can represent whatever its exponents, or an instance feature outside those three parameters. Distinguishing the two would take a tightness census at fixed (n, d) finer than the eighteen values of the calibration campaign, which requires no QPU access and which Section 5.6 records accordingly.

5.3 The chain-break gate as an evaluative metric

The 378 grid cells of Section 4.4.1 support opposite conclusions about κ_2 according to whether the chain-break gate is applied. Unfiltered, Pct_Opt climbs from 42.7% to 81.0% as κ_2 rises from 0.10 to 0.60, a gain of thirty-eight points that would call for a second transfer rule. Filtered, the same cells peak at $\kappa_2 = 0.30$ and decline thereafter, and the count of usable cells falls from 27 of 27 to 6 of 25. The thirty-eight points are produced by samples whose chains have broken, which carry no assignment of the logical variables at all. A quality metric read at face value therefore points the wrong way here, and nothing internal to the objective values reveals it. The correction comes from a hardware diagnostic that the run already records and that no statistical treatment of the objective can substitute for. For benchmarking on annealing hardware it follows that an objective metric should be reported conditioned on a reliability criterion rather than alone, and that the criterion should be stated, since 0.1 is a working choice and Section 2.3.5 records that the literature fixes no threshold.

Two features of the simulated setting bear on how calibration work carried out there should be read, and Ohno et al. (2024) exhibit both. Nothing there has an analogue of the chain-break fraction, which is no defect, simulated annealing having no chains, and it means a law validated in that setting carries no guarantee against the failure mode documented here. Nor does a classical repair layer have any analogue on a QPU sampling a QUBO directly, so a coefficient fitted with repair in place transfers only to a pipeline that supplies one. Both conditions are stated in the sources; neither survives the passage to the hardware, and a rule is portable only as far as the apparatus it was fitted with.

5.4 The calibration architecture as a general strategy

Hyperparameter transfer across problem scales is not peculiar to penalty calibration. Yang et al. (2022) show that under a suitable parametrization of a neural network the optimal hyperparameters become invariant to model width, so that a sweep run cheaply on a small model transfers to a large one with no scaling law at all. Ohno et al. (2024) claim something of that order for the QKP, $c_n \approx 0.09$ making the normalized penalty very nearly invariant to instance size, and Section 4.2 contradicts it: under the normalization adopted here the optimal penalty grows close to linearly with n , and whether some other normalization would recover invariance is a question this thesis leaves open.

The protocol survives the comparison where the invariance does not. A classical exact oracle, cheap at small n , characterizes the dependence of κ_1 on (n, d, α) , and that dependence, imperfect as it is, places a narrow window in a regime whose direct search on the QPU costs more than the budget allows. Invariance would permit transporting a point; its absence permits transporting a window, the same architecture degraded by the residual the regression carries. The window earns its width, the prediction erring in both directions across the six mapped instances (Section 4.4.3), so no constant correction replaces it and $\pm 2\sigma_{\log}$ covers the spread. The classical computation

runs offline on a different and smaller set of instances, only the fitted functional form reaching the problem the QPU is given, which separates it from the hybrid solvers of Section 2.5.4, whose two halves act on one instance inside a single solve.

5.5 Limitations

The two-dimensional grid underlying Protocol P3 was run at a single density. Section 4.4.4 records that the regression-predicted κ_1 returned nothing feasible at all at $d \in \{0.5, 0.75\}$ instead of degrading gradually, so the calibration established here is a $d = 25\%$ result and extending it re-examines the regression, it does not repeat the procedure. The regression itself is fitted at $n \in \{10, 20\}$ and applied at $n = 100$, a factor-five extrapolation, and the paired calibration of Section 4.4.3 shifts the weight of the resulting offset onto extrapolation, away from divergence between the classical and hardware criteria on the evidence of one instance at one size, which resolves the ambiguity only in part.

The grid maps six of the ten `jeu_100_25_*` instances. All ten reach a viable configuration, four through the anchored campaigns alone, so the remaining four lack a map of their viable region and not a working setting, and the claim of Section 4.4.2 that the pivot's region covers the others rests on the six that were mapped. The systematic hardware comparison targets Zephyr, the Pegasus campaign of Section 4.3.3 sampling its cells once against ten elsewhere, with Pegasus viability under the unbalanced formulation resting on four runs at one parameter point. Access to the QPU, not the exhaustion of open questions, fixed all of this — the six-instance grid, the single density, and the coverage of `jeu_100_25_6` and `_8` at fifteen cells against sixty-three elsewhere trace to the cutoff of 15 June 2026.

The early methodological decisions were taken on the pivot before the systematic campaigns were designed, among them the choice to fix κ_2 and the anchor $\kappa_1 = 13$ of Section 3.4.3. The value $\kappa_2 = 0.25$ at which the Billionnet campaign is calibrated comes from manual tuning on `jeu_100_25_1` at $n = 100$, so (16) is fitted at a level the target instance set. The exponent it returns is accordingly that of κ_1 at that level, and Section 4.2 bounds what the level costs by refitting at $\kappa_2 = 0.1$ on the synthetic campaign.

5.6 Future work

Three experiments bear directly on the limitations above, two of them requiring renewed QPU access and one not.

A paired calibration on the same instances against both criteria, run at a size where the exact oracle remains tractable, would separate the two sources of the offset of Section 4.4.3. The single instance reported there points toward extrapolation; a panel at $n \in \{10, 15, 20\}$ with ten seeds per cell, calibrated once against the bitmask oracle and once against QPU samples, would establish whether the two criteria agree in general or only on that instance. This is the experiment that would most directly settle whether the growth of Section 5.1 carries a hardware

component beyond its combinatorial floor.

A grid at $d \in \{0.5, 0.75, 1.0\}$ would establish whether the calibration extends beyond $d = 25\%$. Section 5.5 records that the regression-predicted κ_1 returned nothing feasible at the two intermediate densities, which is a failure abrupt enough to suggest that the multiplicative form itself needs an interaction between size and density rather than a recalibrated exponent.

The third requires no hardware. Section 5.2 shows the calibrated κ_1 rising with tightness before falling, which no single exponent on α can represent, and Section 5.1 rules out the number of selected items as the missing variable. A census over several hundred values of α at fixed (n, d) , calibrated classically, would map the shape of that dependence and show whether a maximum is genuinely present or whether the six instances of Table 6 are showing noise. The classical campaigns already contain the machinery; only the grid of α needs widening.

One further question the present data raises without answering concerns transfer downwards. The setting that reaches the optimum at $n = 100$ returns 28.0% at $n = 10$ and 17.2% at $n = 20$, with chains intact throughout (Section 4.2), so the viable window moves substantially between these sizes. Where it moves, and whether it moves continuously, is measurable against the exact oracle over the whole range where enumeration holds.

6 Conclusion

This thesis asked whether penalty calibration for the QKP, as the simulation-based literature portrays it, survives the transition to a physical quantum processor. It does not survive unchanged, and its failures are more informative than a simple negative would be.

The size exponent goes first, and it goes before any hardware is involved. Re-estimated on 1 437 calibrations against a classical exact oracle, it measures 0.972 (s.e. 0.034) against the 0.09 Ohno et al. (2024) report, a disagreement of an order of magnitude that a t of 25.8 leaves no room to attribute to sampling. The marginal-balance argument of Section 5.1 accounts for it and predicts $c_n = 1$ for both the linear and the squared penalty form, which is what makes the two figures commensurable at all. The estimand separates them. The coefficient they regress is optimal for simulated annealing under a repair procedure that restores feasibility downstream of the penalty, and their own distinction between a critical and an optimal coefficient anticipates that the two decouple. The exponent measured here governs the enforcement floor, the only quantity available to a processor receiving the QUBO intact. A rule fitted below that floor understates it by an order of magnitude at $n = 100$, and three QPU campaigns at $n = 10, 20$ and 100 are consistent with growth of the order measured, the pair reaching the optimum at full size returning 28.0% and 17.2% at the two smaller ones with chains intact throughout.

Across 311 runs of the slack encoding, over two topologies, two embedders, three chain-strength policies and penalty coefficients spanning two orders of magnitude, not one configuration returned a solution at once feasible and chain-reliable. The obstruction has two levels, a clique ceiling of K_{106} against the 107 to 112 variables the encoding needs on Zephyr, and on Pegasus,

whose clique accommodates the problem with margin, a penalty sweep in which the chains break before the returned weight reaches capacity. Formulation separates the working configurations from the failing ones; topology does not. Adopting the slack-free encoding is therefore a material necessity at this size and not a modelling preference, which is the sense in which Chapter 4 sharpens the simulation result of Bontekoe et al. (2023).

Transfer between instances works, in a form weaker than the one claimed. A single calibrated pair applied unchanged to six instances of the same class overruns capacity on two and fills the knapsack to eight per cent on a third, so no point generalizes; the viable regions do overlap, and that overlap is what makes a calibration campaign at $n = 100$ affordable. Transfer from a classical criterion to hardware holds wherever both can be applied to the same problem, the two judges selecting the same κ_1 at $n = 10$. At $n = 100$ the regression misses the QPU optimum by a factor between 0.846 and 1.604 across the six mapped instances, in both directions, enough to place a search window and not enough to serve as a point predictor. Density stops the calibration outright. The regression-predicted penalty returns no feasible solution over 108 cells at $d \in \{0.50, 0.75\}$, with chain-break fractions an order of magnitude below the gate, and at full density the only rule securing feasibility does so by making the empty knapsack the cheapest configuration available. What is established here is a $d = 25\%$ result.

Set against those failures, the benchmark is nonetheless solved. Every one of the ten `jeu_100_25_*` instances admits a penalty setting under which the Advantage2 processor returns a feasible solution with reliable chains, and eight reach the proven optimum of Billionnet and Soutif (2004) exactly. That it holds at $n = 100$ matters for the applications the problem was formulated for, portfolio selection over interdependent risky investments under a budget ceiling (Laughunn, 1970) operating at the hundred-asset scale the benchmark represents. Prediction and search divided the work, the fitted rule locating a window a factor 3.6 wide against the three orders of magnitude an uninformed sweep would span, and 3 555 grid submissions resolving what remained inside it. Whether a functional form exists that would close that residual is a question the present data leaves open, Section 5.2 identifying a non-monotone dependence on tightness and an interaction between size and density as the first candidates.

One methodological finding outlives the QKP. The same 378 grid cells support opposite conclusions about the quadratic penalty coefficient depending on whether a chain-break reliability criterion is applied, an apparent gain of thirty-eight percentage points turning out to come from samples carrying no assignment of the logical variables at all. A quality metric read alone points the wrong way here, and only a hardware diagnostic already recorded in the run corrects it. Benchmark results on annealing hardware should be reported conditioned on such a criterion, and the criterion should be stated, no threshold being established in the literature.

No claim of advantage is made or implied. The Lagrangian decomposition of Billionnet and Soutif (2004) proves optimality on these same instances in about four minutes, and the campaigns reported here consume orders of magnitude more wall-clock time to recover eight of those optima without proving any of them. The hardware serves the measurement, penalty

calibration being observable only once the QUBO reaches a physical sampler intact.

Section 5.5 sets out what bounds these findings, and Section 5.6 what a renewed allocation would settle first. The single experiment that would most directly extend the work is a paired calibration on a panel of instances at a size the exact oracle still reaches, run once against that oracle and once against the QPU. It would establish whether the two criteria agree in general or only on the one instance compared here, and with it whether the near-linear growth measured on the formulation carries a hardware component beyond its combinatorial floor.

7 Bibliography

- Asawalertsak, W., Prechaprapranwong, P., & Sarochawikasis, R. (2025). A Comparative study of D-Wave's Quantum Hybrid Solvers: BQM vs CQM on Combinatorial Optimization. *2025 22nd International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 30–36. <https://doi.org/10.1109/JCSSE67377.2025.11297943>
- Beliakov, G. (2022). Knapsack problems with dependencies through non-additive measures and Choquet integral. *European Journal of Operational Research*, 301(1), 277–286. <https://doi.org/10.1016/j.ejor.2021.11.004>
- Bennett, C. H., Bernstein, E., Brassard, G., & Vazirani, U. (1997). Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5), 1510–1523. <https://doi.org/10.1137/S0097539796300933>
- Billionnet, A., & Calmels, F. (1996). Linear programming for the 0–1 quadratic knapsack problem. *European Journal of Operational Research*, 92(2), 310–325. [https://doi.org/10.1016/0377-2217\(94\)00229-0](https://doi.org/10.1016/0377-2217(94)00229-0)
- Billionnet, A., & Soutif, É. (2004). An exact method based on Lagrangian decomposition for the 0–1 quadratic knapsack problem. *European Journal of Operational Research*, 157(3), 565–575. [https://doi.org/10.1016/S0377-2217\(03\)00244-3](https://doi.org/10.1016/S0377-2217(03)00244-3)
- Bontekoe, T., Phillipson, F., & Schoot, W. van der. (2023). Translating Constraints into QUBOs for the Quadratic Knapsack Problem. In J. Mikyška, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, & P. M. Slood (Eds.), *Computational Science – ICCS 2023* (pp. 90–107). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-36030-5_8
- Boothby, K., Bunyk, P., Raymond, J., & Roy, A. (2019). Next-Generation Topology of D-Wave Quantum Processors. https://www.dwavequantum.com/media/jwwj5z3z/14-1026a-c_next-generation-topology-of-dw-quantum-processors.pdf
- Boothby, K., King, A. D., & Raymond, J. (2021). Zephyr Topology of D-Wave Quantum Processors. https://www.dwavequantum.com/media/2uznec4s/14-1056a-a_zephyr_topology_of_d-wave_quantum_processors.pdf
- Buonaiuto, G., Gargiulo, F., De Pietro, G., Esposito, M., & Pota, M. (2023). Best practices for portfolio optimization by quantum computing, experimented on real quantum devices. *Scientific Reports*, 13. <https://doi.org/10.1038/s41598-023-45392-w>
- Cacchiani, V., Iori, M., Locatelli, A., & Martello, S. (2022). Knapsack problems — An overview of recent advances. Part I: Single knapsack problems. *Computers & Operations Research*, 143, 105692. <https://doi.org/10.1016/j.cor.2021.105692>
- Caprara, A., Pisinger, D., & Toth, P. (1999). Exact Solution of the Quadratic Knapsack Problem. *INFORMS Journal on Computing*, 11(2), 125–137. <https://doi.org/10.1287/ijoc.11.2.125>

- Chaillou, P., Hansen, P., & Mahieu, Y. (1989). Best network flow bounds for the quadratic knapsack problem. In B. Simeone (Ed.), *Combinatorial Optimization* (pp. 225–235). Springer. <https://doi.org/10.1007/BFb0083467>
- Chen, Y., & Hao, J.-K. (2017). An iterated “hyperplane exploration” approach for the quadratic knapsack problem. *Computers & Operations Research*, 77, 226–239. <https://doi.org/10.1016/j.cor.2016.08.006>
- Creemers, S. (2025). Speeding up Grover’s algorithm. *European Journal of Operational Research*, 326(1), 13–27. <https://doi.org/10.1016/j.ejor.2025.02.034>
- Creemers, S., & Armas, L. F. P. (2025). Discrete optimization: A quantum revolution? *European Journal of Operational Research*, 323(2), 378–408. <https://doi.org/10.1016/j.ejor.2024.12.016>
- Dantzig, G. B. (1957). Discrete-Variable Extremum Problems. *Operations Research*, 5(2), 266–288. <https://doi.org/10.1287/opre.5.2.266>
- Deleplanque, S., Durbec, A., & El Yaagoubi, A. (2025). Optimizing bin packing problem on quantum annealers: A QUBO-based approach and scalability challenges. *Optimization and Engineering*. <https://doi.org/10.1007/s11081-025-10047-4>
- Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2000, January 28). *Quantum Computation by Adiabatic Evolution*. arXiv: [quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106). <https://doi.org/10.48550/arXiv.quant-ph/0001106>
- Fennich, M. E., Fomeni, F. D., & Coelho, L. C. (2024). A novel dynamic programming heuristic for the quadratic knapsack problem. *European Journal of Operational Research*, 319(1), 102–120. <https://doi.org/10.1016/j.ejor.2024.06.034>
- Fomeni, F. D., & Letchford, A. N. (2014). A Dynamic Programming Heuristic for the Quadratic Knapsack Problem. *INFORMS Journal on Computing*, 26(1), 173–182. <https://doi.org/10.1287/ijoc.2013.0555>
- Fortet, R. (1960). L’algebre de Boole et ses applications en recherche operationnelle. *Trabajos de Estadística*, 11(2), 111–118. <https://doi.org/10.1007/BF03006558>
- Galli, L., Martello, S., Rey, C., & Toth, P. (2021). Polynomial-size formulations and relaxations for the quadratic multiple knapsack problem. *European Journal of Operational Research*, 291(3), 871–882. <https://doi.org/10.1016/j.ejor.2020.10.047>
- Galli, L., Martello, S., & Toth, P. (2025). The quadratic knapsack problem. *European Journal of Operational Research*, 326(1), 1–12. <https://doi.org/10.1016/j.ejor.2024.12.032>
- Gallo, G., Hammer, P. L., & Simeone, B. (1980). Quadratic knapsack problems. In M. W. Padberg (Ed.), *Combinatorial Optimization* (pp. 132–149, Vol. 12). Springer Berlin Heidelberg. <https://doi.org/10.1007/BFb0120892>
- Garey, M., & Johnson, D. (1978). “Strong” NP-Completeness Results: Motivation, Examples, and Implications. *Journal of the ACM (JACM)*, 25(3), 499–508. <https://doi.org/10.1145/322077.322090>

- Glover, F., Kochenberger, G., & Alidaee, B. (2002). Solving quadratic knapsack problems by reformulation and tabu search: single constraint case. *Combinatorial and Global Optimization*, 14. https://doi.org/10.1142/9789812778215_0008
- Grant, E., Humble, T. S., & Stump, B. (2021). Benchmarking Quantum Annealing Controls with Portfolio Optimization. *Physical Review Applied*, 15(1), 014012. <https://doi.org/10.1103/PhysRevApplied.15.014012>
- Grover, L. K. (1996, November 19). A fast quantum mechanical algorithm for database search. arXiv: [quant-ph/9605043](https://arxiv.org/abs/quant-ph/9605043). <https://doi.org/10.48550/arXiv.quant-ph/9605043>
- Güney, E., Ehrenthal, J., & Hanne, T. (2025). QUBO Formulations and Characterization of Penalty Parameters for the Multi-Knapsack Problem. *IEEE Access*, 13, 47086–47098. <https://doi.org/10.1109/ACCESS.2025.3550788>
- Hammer, P., & Rader Jr., D. J. (1997). Efficient Methods For Solving Quadratic 0–1 Knapsack Problems. *INFOR: Information Systems and Operational Research*, 35(3), 170–182. <https://doi.org/10.1080/03155986.1997.11732327>
- Helmberg, C. (2000). Fixing Variables in Semidefinite Relaxations. *SIAM Journal on Matrix Analysis and Applications*, 21(3), 952–969. <https://doi.org/10.1137/S089547989631442X>
- Jeong, S.-G., Cong, M. D., Noh, D.-I., Pham, Q.-V., & Hwang, W.-J. (2025). Embedding-Aware Noise Modeling of Quantum Annealing (2). <https://doi.org/10.48550/ARXIV.2510.04594>
- Kadowaki, T., & Nishimori, H. (1998). Quantum Annealing in the Transverse Ising Model. *Physical Review E*, 58(5), 5355–5363. <https://doi.org/10.1103/PhysRevE.58.5355>
- Karp, R. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, 40, 85–103. https://doi.org/10.1007/978-3-540-68279-0_8
- Khot, S. (2004). Ruling out PTAS for graph min-bisection, densest subgraph and bipartite clique. *45th Annual IEEE Symposium on Foundations of Computer Science*, 136–145. <https://doi.org/10.1109/FOCS.2004.59>
- Laughunn, D. J. (1970). Quadratic Binary Programming with Application to Capital-Budgeting Problems. *Operations Research*, 18(3), 454–461. <https://doi.org/10.1287/opre.18.3.454>
- Lucas, A. (2014). Ising formulations of many NP problems. *Frontiers in Physics*, 2. <https://doi.org/10.3389/fphy.2014.00005>
- Martello, S., & Toth, P. (1990). Knapsack problems: algorithms and computer implementations. <https://doi.org/10.2307/2583458>
- Montañez-Barrera, A., Willsch, D., Maldonado-Romo, A., & Michielsen, K. (2024). Unbalanced penalization: A new approach to encode inequality constraints of combinatorial problems for quantum optimization algorithms. *Quantum Science and Technology*, 9(2), 025022. <https://doi.org/10.1088/2058-9565/ad35e4>
- Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4), 308–313. <https://doi.org/10.1093/comjnl/7.4.308>

- O’Gorman, B., Fluegemann, J., Biswas, R., & Smelyanskiy, V. (2015). Determination and correction of persistent biases in quantum annealers. *Scientific Reports*, 6. <https://doi.org/10.1038/srep18628>
- Ohno, K., Shirai, T., & Togawa, N. (2024). Toward Practical Benchmarks of Ising Machines: A Case Study on the Quadratic Knapsack Problem. *IEEE Access*, 12, 97678–97690. <https://doi.org/10.1109/ACCESS.2024.3425711>
- Parizy, M., & Togawa, N. (2021). Analysis and Acceleration of the Quadratic Knapsack Problem on an Ising Machine. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E104.A(11), 1526–1535. <https://doi.org/10.1587/transfun.2020KEP0007>
- Park, H., & Lee, H. (2024). Determination of optimal chain coupling made by embedding in d-wave quantum annealer. *AVS Quantum Science*. <https://doi.org/10.1116/5.0205511>
- Pelofske, E. (2023). Comparing three generations of d-wave quantum annealers for minor embedded combinatorial optimization problems. *Quantum Science and Technology*, 10. <https://doi.org/10.1088/2058-9565/adb029>
- Pérez Armas, L. F., Creemers, S., & Deleplanque, S. (2024). Solving the resource constrained project scheduling problem with quantum annealing. *Scientific Reports*, 14(1), 16784. <https://doi.org/10.1038/s41598-024-67168-6>
- Pisinger, D. (2005). Where are the hard knapsack problems? *Comput. Oper. Res.*, 32, 2271–2284. <https://doi.org/10.1016/j.cor.2004.03.002>
- Pisinger, D. (2007). The quadratic knapsack problem - a survey. *Discret. Appl. Math.*, 155, 623–648. <https://doi.org/10.1016/j.dam.2006.08.007>
- Preskill, J. (2018). Quantum Computing in the NISQ era and beyond. *Quantum*, 2, 79. <https://doi.org/10.22331/q-2018-08-06-79>
- Pusey-Nazzaro, L., & Date, P. (2020, August 17). *Adiabatic Quantum Optimization Fails to Solve the Knapsack Problem*. arXiv: 2008.07456 [quant-ph]. <https://doi.org/10.48550/arXiv.2008.07456>
- Qpu solver parameters* [Ocean SDK documentation, v9.4.0]. (n.d.). https://docs.dwavequantum.com/en/latest/quantum_research/solver_parameters.html
- Rader Jr., D. J., & Woeginger, G. J. (2002). The quadratic 0–1 knapsack problem with series–parallel support. *Operations Research Letters*, 30(3), 159–166. [https://doi.org/10.1016/S0167-6377\(02\)00122-0](https://doi.org/10.1016/S0167-6377(02)00122-0)
- Reference Documentation — D-Wave Quantum Computing Products documentation* [Ocean SDK documentation, v9.4.0]. (n.d.). <https://docs.dwavequantum.com/en/latest/ocean/packages.html#ocean-packages>
- Rhys, J. M. W. (1970). A Selection Problem of Shared Fixed Costs and Network Flows. *Management Science*, 17(3), 200–207. <https://doi.org/10.1287/mnsc.17.3.200>
- Rosenberg, G., Haghnegahdar, P., Goddard, P., Carr, P., Wu, K., & de Prado, M. L. (2016). Solving the Optimal Trading Trajectory Problem Using a Quantum Annealer. *IEEE*

- Journal of Selected Topics in Signal Processing*, 10(6), 1053–1060. <https://doi.org/10.1109/JSTSP.2016.2574703>
- Tikhonov, A. (1963). Solution of incorrectly formulated problems and the regularization method. 5, 1035–1038.
- Topologies — D-Wave Quantum Computing Products documentation* [Ocean SDK documentation, v9.4.0]. (n.d.). https://docs.dwavequantum.com/en/latest/quantum_research/topologies.html
- Venturelli, D., & Kondratyev, A. (2019). Reverse Quantum Annealing Approach to Portfolio Optimization Problems. *Quantum Machine Intelligence*, 1(1–2), 17–30. <https://doi.org/10.1007/s42484-019-00001-w>
- Watters, L. J. (1967). Reduction of Integer Polynomial Programming Problems to Zero-One Linear Programming Problems. *Operations Research*, 15(6), 1171–1174. <https://www.jstor.org/stable/168623>
- Yang, G., Hu, E. J., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., & Gao, J. (2022, March 28). Tensor programs v: tuning large neural networks via zero-shot hyperparameter transfer. <https://doi.org/10.48550/arXiv.2203.03466>
- Yarkoni, S., Raponi, E., Bäck, T., & Schmitt, S. (2022). Quantum annealing for industry applications: introduction and review. *Reports on Progress in Physics*, 85(10), 104001. <https://doi.org/10.1088/1361-6633/ac8c54>

A Appendix

A.1 Connectivity and embedding of the QUBO-encoded QKP

A.1.1 Density-independent connectivity of the penalty term

Section 2.3.4 and Section 2.4 both invoke the fact that the QUBO reformulation of the QKP's capacity constraint couples every pair of items, regardless of the objective's density d .

Both the slack penalty (9) and the unbalanced penalty (11) contain, as their quadratic term, a multiple of $(\sum_i w_i x_i)^2$, with or without slack bits added to the sum, whose presence does not affect the argument below. Expanding, and using $x_i^2 = x_i$ for binary variables,

$$\left(\sum_i w_i x_i\right)^2 = \sum_i w_i^2 x_i + 2 \sum_{i < j} w_i w_j x_i x_j. \quad (20)$$

The coefficient of $x_i x_j$ in (20) is strictly positive whenever both weights are, which holds for every pair in all instances used in this thesis, weights being drawn strictly positive under every generation protocol employed (Section 2.1.3). The resulting coupling is independent of d , which governs only the sparsity of the separate term $\sum_{i < j} p_{ij} x_i x_j$ in the objective (3). Lucas (2014) make the same observation in general for inequality-constrained Ising formulations.

The argument would fail only if a penalty coupling cancelled the corresponding objective coefficient exactly, which requires $p_{ij} = 2\lambda_2 w_i w_j$ (or $p_{ij} = 2\kappa_2 w_i w_j P_{\max}/W_{\max}^2$ following Section 3.2.2 convention) for some pair. Profits and weights being drawn independently over the integers, no instance used here produces that coincidence.

Take $w = (1, 2, 3)$ with $C = 4$, and an objective carrying a single pairwise profit, on the pair (x_1, x_2) , so that its interaction graph has one edge of the three available and $d = 1/3$. The slack encoding adds three bits, and expanding the squared deviation produces a coupling $2\lambda w_i w_j$ on every pair, worth 60 on (x_1, x_3) and 120 on (x_2, x_3) at $\lambda = 10$, neither of which carries any objective coefficient. On (x_1, x_2) , where a profit of 3 enters the minimization as -3 and the penalty contributes $+40$, the final coefficient is $+37$: the two terms add on that edge rather than replacing one another, which is the sense in which the caveat above binds. Figure 3 shows the two graphs, the objective with one edge and the QUBO with all fifteen.

The clique-size ceiling of Section 2.3.4 therefore applies at every density and not only in the dense case. The QUBO graph is complete on the n item variables regardless of d , and the slack formulation extends the clique further, to $n + \lceil \log_2(C + 1) \rceil$ nodes, by coupling every item to every slack bit.

A.1.2 The chain Hamiltonian of minor embedding

Section A.1.1 shows that every item pair is coupled regardless of density, and Section 2.3.5 defines the chain-break fraction informally on that basis. The explicit Hamiltonian below locates

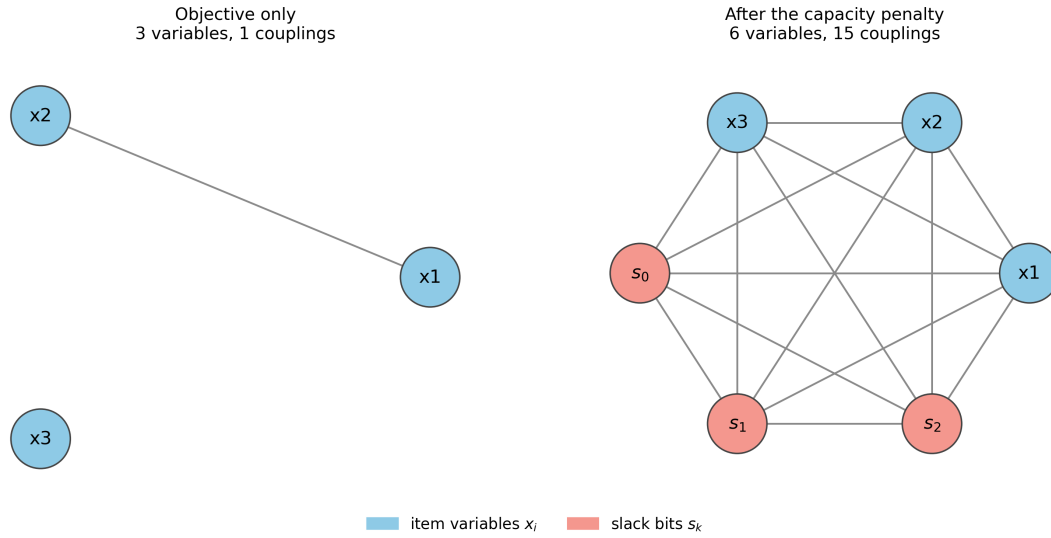


Figure 3 – Interaction graphs for the three-item instance $w = (1, 2, 3)$, $C = 4$. Left: the objective at $d = 1/3$, carrying its single pairwise profit on (x_1, x_2) . Right: the QUBO after the slack penalty, complete on the three item variables and the three slack bits.

where chain breaks originate, following Venturelli and Kondratyev (2019).

Let a logical Ising problem be given by linear biases h_i and couplings J_{ij} over N logical variables, and let $\chi = (V, E)$ denote the physical hardware graph, Chimera in the original derivation and Pegasus or Zephyr in this thesis. If $(i, j) \notin \chi$, the coupling J_{ij} cannot be realized directly, couplers existing only between qubits close enough in the processor's layout. Minor embedding resolves this by representing each logical variable i with a chain of N_c physical qubits, indexed σ_{ic}^z for $c = 1, \dots, N_c$, giving

$$H_{\chi\text{-Ising}} = - \sum_{i=1}^N |J_F| \left[\sum_{c=1}^{N_c-1} \sigma_{ic}^z \sigma_{i(c+1)}^z \right] \quad (21)$$

$$+ \sum_{i=1}^N \frac{h_i}{N_c} \left[\sum_{c=1}^{N_c} \sigma_{ic}^z \right] + \sum_{i,j=1}^N J_{ij} \sum_{c_i, c_j=1}^{N_c} \delta_{ij}^{\chi}(c_i, c_j) \sigma_{ic_i}^z \sigma_{jc_j}^z, \quad (22)$$

where σ_{ic}^z is the c -th physical qubit of the chain representing logical variable i ; J_F is the chain strength, a ferromagnetic coupling applied uniformly along each chain in (21); h_i and J_{ij} are the original logical bias and coupling, redistributed in (22) over the chain with the local field split evenly across its N_c qubits; and $\delta_{ij}^{\chi}(c_i, c_j)$ is an adjacency indicator equal to one only for the single pair of physical qubits, one in each chain, that a physical coupler connects, which confines each logical coupling J_{ij} to one edge of χ .

Term (21) is the source of the chain-break phenomenon of Section 2.3.5. For the chain to behave as a single effective spin, J_F must be strong enough that $\langle \sigma_{ic}^z \rangle = \langle \sigma_{ic'}^z \rangle$ for all c, c' within it (Venturelli and Kondratyev, 2019). Term (22) carries the optimization problem itself, redistributed over the chain through δ_{ij}^{χ} , and it is this term that competes with (21) for the same

finite coupler precision, a competition a weak J_F can lose.

The chain length N_c is a property of the embedding rather than of the Hamiltonian. In the derivation of Venturelli and Kondratyev (2019) it follows from the Chimera topology of the machine they use, where a complete graph on N logical variables embeds with $N_c = N/4 + 1$. That relation does not carry to Pegasus or Zephyr, whose chain lengths are those Section 4.5 reports from busclique.

A.2 Complete penalty sweep, slack formulation on Pegasus

The campaign summarized in Section 4.3.3 swept the penalty coefficient of the slack formulation over 65 distinct values on the pivot instance `jeu_100_25_1`, using busclique embedding and `uniform_torque_compensation`. The capacity is 669: every row below exceeds it, so **no configuration in this sweep is feasible**. Pct_Opt is reported for completeness but carries no meaning here, since an infeasible configuration has no interpretable objective value; values above 100% arise from configurations that violate the capacity constraint outright. Values are averaged over repeated reads where a coefficient was sampled more than once.

λ	Pct_Opt (%)	Weight	cbf	λ	Pct_Opt (%)	Weight	cbf
0.0	354.4	2582	0.000	3.3	80.8	1061	0.446
0.1	354.4	2582	0.009	3.4	72.8	1044	0.452
0.2	299.9	2085	0.107	3.5	77.7	1028	0.453
0.3	243.1	1687	0.225	3.6	67.4	1055	0.451
0.4	198.6	1407	0.309	3.7	79.4	1075	0.449
0.5	166.6	1217	0.354	3.8	74.0	1051	0.453
0.6	156.4	1226	0.393	3.9	74.5	985	0.456
0.7	143.1	1189	0.406	4.0	74.8	990	0.420
0.8	118.5	1111	0.423	5.0	57.8	985	0.374
0.9	125.5	1164	0.432	6.0	62.7	1008	0.424
1.0	118.5	1133	0.402	8.0	58.2	1017	0.418
1.1	105.3	1082	0.437	10.0	55.0	982	0.398
1.2	102.6	1086	0.444	12.0	59.9	995	0.428
1.3	108.7	1151	0.442	14.0	61.1	982	0.424
1.4	97.2	1077	0.447	16.0	55.3	1001	0.420
1.5	90.0	1022	0.443	18.0	59.5	1010	0.418
1.6	90.0	1041	0.441	20.0	57.6	990	0.407
1.7	95.6	1074	0.443	22.0	54.6	986	0.414
1.8	81.5	1057	0.446	24.0	55.2	1006	0.415
1.9	84.1	1031	0.445	26.0	58.4	981	0.418
2.0	74.5	998	0.424	28.0	57.9	980	0.420
2.1	90.4	1094	0.440	30.0	59.0	980	0.418
2.2	77.0	1017	0.443	32.0	59.5	998	0.421
2.3	86.4	1043	0.449	34.0	55.5	1000	0.418
2.4	80.3	1032	0.453	36.0	60.6	996	0.416
2.5	89.9	1101	0.449	38.0	55.2	989	0.417
2.6	77.9	1029	0.451	40.0	54.9	986	0.416
2.7	83.8	1071	0.449	42.0	54.1	982	0.420
2.8	78.1	1071	0.455	44.0	59.5	983	0.418
2.9	67.5	992	0.450	46.0	50.9	961	0.421
3.0	72.5	1011	0.409	48.0	57.5	981	0.416
3.1	89.2	1087	0.451	50.0	50.4	970	0.378
3.2	70.4	1052	0.450				

Feasible column omitted: it is uniformly “Nan” across all 65 rows.

A.3 Instance generators

The two generated sets of Section 3.1.2 draw their coefficients as follows.

The synthetic generator assigns a linear profit to every item, drawn uniformly in $[1, 100]$, a pairwise profit in the same range with probability d , and weights uniformly in $[1, 100]$. Its capacity is $\max(\lfloor \alpha \sum_i w_i \rfloor, \max_i w_i)$, the floor guaranteeing that at least the heaviest item fits.

The Billionnet generator draws profits, linear and pairwise alike, uniformly in $[0, 100]$ subject to the density, so that a fraction $1 - d$ of items carries no individual profit. Weights are uniform in $[1, 50]$ and the capacity is $\max(50, \lfloor \alpha \sum_i w_i \rfloor)$, the floor of 50 being that of the published protocol (Billionnet and Soutif, 2004).

A.4 Pipeline verification

Two normalization anomalies were identified and rectified prior to the calibration campaigns. In a preliminary implementation, only the objective was divided by $P_{\max}$. Consequently, the weights entered the penalty at their raw scale, inflating the penalty couplers by a factor of $W_{\max}$ relative to the objective and undermining the purpose of the dimensionless parameter κ . Furthermore, normalizing both the weights and the penalty coefficient simultaneously applied the division twice. The adopted convention divides the weights exactly once during construction and treats κ as inherently dimensionless, precluding any further downstream scaling. These corrections were numerically validated before initiating the campaigns.

The exact oracle was independently re-executed across the entire Billionnet dataset when revisiting the calibrated values in Section 5.1. Each instance was regenerated from its seed, and its configuration space was re-enumerated via a distinct code path. The 1437 recomputed optima exactly match those recorded during the initial calibration.

The substantial volume of QPU records necessitated two operational safeguards. Given that D-Wave SampleSet objects retain the full physical state of every read—comprising 1 149 qubits across 2 000 reads per call at $n = 100$ —each sampling operation is followed by an explicit memory release of both the sample set and the model. Additionally, campaign results are appended to disk instance by instance rather than accumulated in memory, ensuring that completed computations are preserved in the event of an interruption.

Three parameters of the Nelder-Mead calibration were evaluated prior to execution. On the tested instances, convergence is systematically achieved within 150 iterations, remaining well below the imposed threshold of 300. Sweeping the regularization coefficient across $\rho \in [10^{-4}, 10]$ maintains the gap_energy at zero up to $\rho = 0.1$, before triggering a sharp increase at $\rho = 1$. The retained value thus corresponds to the threshold where the penalty does not artificially inflate the objective energy. Finally, the three initializations distributed along the κ_1 axis consistently converge to an identical minimum wherever a zero-gap interval is present.

A.5 Grid coverage per instance

Execution of the two-dimensional grid departed from the design of Section 3.4.4 as QPU access ran short. Table A.1 records what each instance received.

Table A.1 – Coverage of the two-dimensional grid campaigns. The design calls for 63 cells at ten submissions each; the earliest block used five submissions and a one-sided window.

Instance	Cells	κ_2 levels	κ_1 window	Submissions
jeu_100_25_1	63	7	symmetric, $\pm 2\sigma_{\log}$	10
jeu_100_25_2	63	7	symmetric, $\pm 2\sigma_{\log}$	10
jeu_100_25_3	129	8	symmetric, $\pm 2\sigma_{\log}$	5 and 10
jeu_100_25_4	63	7	symmetric, $\pm 2\sigma_{\log}$	10
jeu_100_25_6	30	5	upper half only	5 and 10
jeu_100_25_8	30	5	upper half only	5 and 10

The earliest block, of 45 cells over three instances, was centred on the preliminary regression rather than on (16), which places its κ_1 nodes about 11% above those of the later campaigns at the same σ multiplier. Three cells at $\kappa_2 = 0.75$ on jeu_100_25_3 lie outside the range Section 3.4.4 describes, two of them viable, at 90.2% and 91.5% of the proven optimum.

A.6 Method A, complete results

Section 4.4.5 reports the best viable configuration reached on each instance. Table A.2 gives the complementary statistic, the mean over all viable cells at each level of κ_2 , which is what a practitioner choosing a single configuration without instance-level information would face.

Table A.2 – Mean Pct_Opt over viable cells, anchored campaigns on the ten jeu_100_25_* instances, by κ_2 and configuration. ref applies $\kappa_1 = 13.0$ unchanged; corr_bill and corr_synth apply the tightness correction of (17) with $c_\alpha = -0.1924$ and -0.1451 .

κ_2	ref	corr_bill	corr_synth
0.10	87.8	81.0	84.3
0.20	89.1	87.9	88.0
0.25	90.5	87.8	88.4
0.40	98.8	80.0	79.9

The $\kappa_2 = 0.25$ level was run twice. The earlier campaign, which does not record the number of feasible repetitions per cell, is excluded from this table and from every figure in Chapter 4. The row at $\kappa_2 = 0.40$ rests on four viable cells out of ten against eight at $\kappa_2 = 0.25$, so its higher mean reports the survivors of a stricter gate.

A.7 Regression diagnostics

$$\kappa_1 = 0.395 \cdot n^{0.97} \cdot d^{1.04} \cdot \alpha^{-0.19} \text{ -- real data, Billionnet campaign (1440 instances)}$$

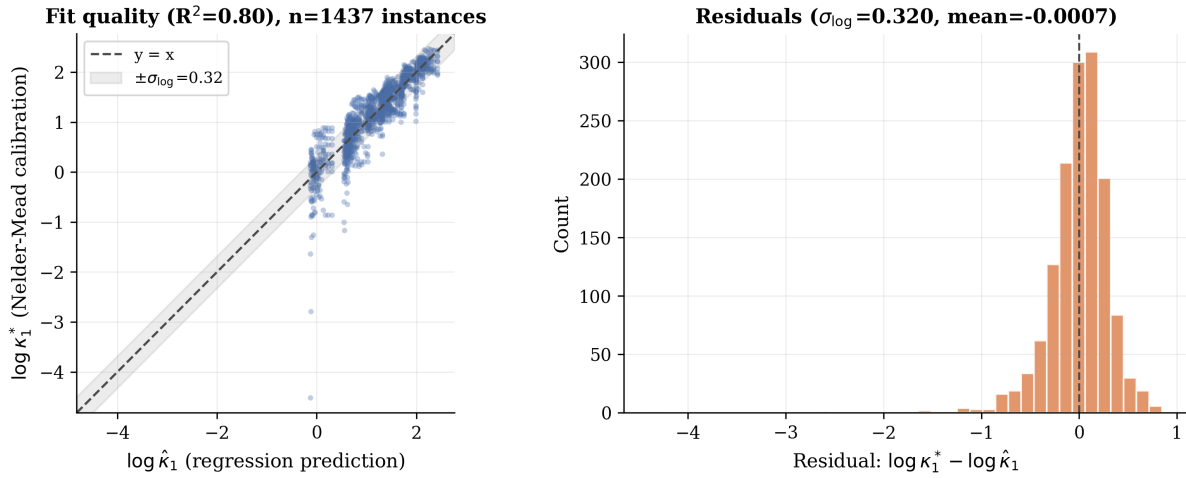


Figure 4 – Log-linear regression (16) on the Billionnet campaign, 1437 instances with $\kappa_1^* > 0$. Left: predicted against observed $\log \kappa_1$, with the $\pm \sigma_{\log}$ envelope. Right: residual distribution, mean -0.0007 , standard deviation 0.3203 , largest residual 4.39 .

The residual distribution is centred, with a standard deviation between 0.10 and 0.39 across most of the campaign. It widens where the constraint approaches inactivity, reaching 0.810 in the cell combining $\alpha > 0.8$ with $d = 0.25$, which also holds the largest residual of the campaign at 4.39 . The effect is present at every density and amplified at the sparsest, a factor 2.5 against 1.4 at $d = 1$. The calibrated κ_1 in that corner averages 1.10 with a minimum of 0.011 , so the model is fitting instances whose penalty has almost nothing left to enforce.

Table A.3 – Log-linear regression of $\log \kappa_1^*$ on instance features, Billionnet-protocol campaign, 1437 observations over 144 design cells. Standard errors are clustered by cell, the ten seeds within a cell sharing identical regressors; a block bootstrap over the cells returns 0.0337 on $\log n$, and the corresponding OLS errors are 0.0244 , 0.0163 and 0.0134 . $A = 0.3947$, $R^2 = 0.8040$, $\sigma_{\log} = 0.3203$; Ohno’s corresponding prefactor is 1.14 .

Feature	Coefficient	s.e.	t (vs. 0)	Ohno	t (vs. Ohno)
$\log n$	$+0.9723$	0.0342	$+28.43$	$+0.09$	$+25.80$
$\log d$	$+1.0416$	0.0260	$+40.02$	$+0.84$	$+7.75$
$\log \alpha$	-0.1924	0.0228	-8.42	-0.21	$+0.77$

A.8 Data and code availability

All computation used Ocean SDK 9.3.0 (dwave-ocean-sdk) under Python 3.13.3. Instance files, calibration and grid records in CSV, QPU sample metadata, and the code that produced

them are archived at [<https://github.com/Nope-Nieks/master-thesis-calibrating-the-Quadratic-Knapsack-Problem-on-physical-hardware.git>], organised as follows.

- `data/instances/` — the Billionnet & Soutif benchmark files used, and the generators for the two calibration sets of Section 3.1.2.
- `src/calibration/` — Nelder-Mead calibration against the bitmask oracle and against QPU samples.
- `src/grid/` — the two-dimensional grid campaigns and the anchored campaigns.
- `results/kappa/` — calibrated κ_1^* per instance, one file per campaign.
- `results/grid/` — the 378 grid cells with feasibility, objective, and chain-break records.
- `analysis/` — the regressions of Section 3.4.2, the unregularized refit, and the scripts producing every figure and table of Chapters 4 and 5.

