Prioritizing Post-Quantum Migration in Synthetic Cryptographic Inventories with Machine Learning and QUBO Optimization

Larissa de Oliveira Figueira[1], Luiz Otavio Duarte[2], Rafael Lucas Silva[3]

[1]Facti – Fundação de Apoio à Capacitação em Tecnologia da Informação
Brasil

{larissa.oliveira,luiz.duarte,rafael.silva}@facti.com.br

Abstract. *Post-quantum cryptography migration requires prioritizing cryptographic assets under budget and dependency constraints. This paper presents a reproducible pipeline combining supervised learning, heuristics, MILP, and QUBO. The evaluation covers three synthetic inventories with 1,609 assets, 1,179 migration candidates, and 1,050 dependencies. Logistic regression was selected for system-grouped priority estimation. Linear MILP achieved the highest predicted priority, closely followed by the priority-to-cost greedy method, while dependency-aware methods produced more coherent portfolios. Adding dependencies to QUBO reduced mean mandatory violations from 10.963 to 7.859 and increased coherence from 0.344 to 0.479, with a small priority decrease. Under the adopted soft objective, soft-dependency MILP achieved the best trade-off and substantially lower runtime. These results show that explicit dependency modeling improves migration coherence and that QUBO provides a flexible representation of budget and dependency interactions.*
Keywords: *post-quantum cryptography; migration prioritization; cryptographic inventories; machine learning; QUBO.*

1. Introduction

Contemporary digital infrastructures depend on public-key cryptography for authentication, digital signatures, certificate validation, key establishment, software updates, and trust management. Widely deployed mechanisms such as RSA, Diffie–Hellman, ECDH, and ECDSA rely on integer-factorization or discrete-logarithm problems, which can be solved in polynomial time by Shor's algorithm on a sufficiently large fault-tolerant quantum computer [Shor 1994]. In response, NIST standardized ML-KEM for key establishment and ML-DSA and SLH-DSA for digital signatures [NIST 2024a, NIST 2024b, NIST 2024c].

Post-quantum cryptography (PQC) migration, however, is not a direct algorithm-replacement process. Cryptographic mechanisms are embedded in applications, protocols, libraries, certificates, firmware, cloud services, legacy systems, and third-party products. Their replacement must preserve interoperability, operational continuity, and chains of trust. The urgency is reinforced by *harvest now, decrypt later* attacks, in which encrypted data collected today may be decrypted once cryptographically relevant quantum computers become available [CISA et al. 2023]. Migration guidance therefore recommends early planning based on cryptographic inventories, risk assessment, dependency identification, supplier engagement, cryptographic agility, and staged execution [CISA et al. 2023, ETSI 2020]. Although inventories provide visibility into quantum-vulnerable assets, they do not determine how limited resources should be allocated among assets with different levels of criticality, exposure, data sensitivity, technical readiness, interoperability risk, and migration cost.

Technical dependencies further complicate this decision. Applications may depend on libraries, certificates, identity services, communication infrastructure, and coordinated software or firmware updates. Previous work has shown that dependency analysis supports the identification and prioritization of cryptographic systems during PQC migration [Hasan et al. 2024]. Related studies have addressed lifecycle trust, hybrid post-quantum signatures, artifact provenance, and PQC controls in software and artificial-intelligence supply chains [Silva 2026, Silva and Duarte 2026b, Silva and Duarte 2026a]. However, these works do not directly address migration-portfolio selection under simultaneous budget and dependency considerations.

This study formulates PQC migration prioritization as a budget-constrained binary selection problem related to the 0/1 knapsack problem [Martello and Toth 1990]. Each candidate asset has a predicted migration-priority score and an estimated cost, while pairwise relationships represent mandatory prerequisites or optional coordination benefits. The proposed pipeline combines supervised machine learning with heuristic, Mixed-Integer Linear Programming (MILP), and Quadratic Unconstrained Binary Optimization (QUBO) policies. System-grouped out-of-fold priority estimates are used as benefits in the portfolio-selection stage. The QUBO formulations represent benefits, budget feasibility, and dependencies through quadratic penalty and reward terms [Glover et al. 2019, Lucas 2014] and are solved using classical simulated annealing, without claims of quantum or computational advantage.

The research hypothesis is that explicitly modeling technical dependencies reduces violations and improves migration-portfolio coherence, although it may moderately reduce mitigated priority. Rather than assuming that QUBO outperforms classical formulations, the study evaluates trade-offs among aggregate benefit, dependency consistency, and computational cost. The evaluation examines whether priority estimates remain stable across independent inventories, whether dependency modeling improves coherence, what costs in priority and runtime it introduces, and how QUBO compares with equivalent heuristic and MILP policies. Three independent generated synthetic inventories are used, totaling 240 systems, 1,609 cryptographic assets, 1,179 migration candidates, and 1,050 technical dependencies. Nine portfolio-selection policies are compared across 135 common scenarios. The contributions are: i) a reproducible pipeline combining synthetic inventory generation, system-grouped priority estimation, and portfolio selection; ii) a comparison of heuristic, MILP, and QUBO policies under alternative dependency treatments; and iii) a multi-inventory evaluation with repeated stochastic executions, paired bootstrap analyses, and publicly available replication artifacts [Figueira et al. 2026].

2. Methodology

This study evaluates a reproducible pipeline for prioritizing PQC migration under budget and dependency constraints. Using synthetic inventories, the pipeline combines supervised priority estimation, independent QUBO calibration, and comparative portfolio optimization. Calibration and evaluation use disjoint inventories to avoid test-set parameter selection.

2.1. Experimental Design and Priority Estimation

The simulator models organizational systems containing certificates, keys, protocols, cryptographic libraries, VPN components, certificate authorities, and software- or firmware-signing artifacts. Asset attributes represent quantum vulnerability, criticality, data sensitivity, retention, exposure, regulatory relevance, PQC readiness, cryptographic agility, migration complexity, interoperability risk, and cost, following guidance on cryptographic discovery, risk assessment, and dependency analysis [CISA et al. 2023, ETSI 2020, Hasan et al. 2024]. Migration costs are expressed in a common monetary unit, while sparse directed graphs model intra- and intersystem dependencies. Each edge (i, j) has weight d_{ij} and is classified as mandatory when asset i requires prerequisite j, or optional when joint migration provides a coordination benefit. A simulator-defined priority score is computed from technical and organizational attributes, with assets in the top 30% of the score distribution defining the high-priority class. These labels are experimental constructs rather than empirical risk estimates.

Logistic Regression, Random Forest, and Gradient Boosting are compared using system-grouped partitions to prevent related assets from crossing data subsets [Hastie et al. 2009]. Numerical features are standardized and categorical features are one-hot encoded. Logistic Regression achieved the best validation performance, while sigmoid and isotonic calibration provided no consistent improvement. Each candidate therefore receives a five-fold system-grouped out-of-fold probability r_i, used as the optimization benefit.

Evaluation uses three inventories generated with seeds 42, 142, and 242, each with 80 systems and 3–10 assets per system, plus three disjoint inventories for QUBO calibration. Candidate sets contained 60, 120, and 180 assets, with budgets of 10%, 20%, and 30% of aggregate candidate cost. Five samples are generated for each inventory and candidate-set size, totaling 135 scenarios. The random baseline is repeated 20 times per scenario, and each QUBO scenario uses 50 independent simulated annealing runs with 2,500 iterations each.

2.2. Portfolio-Selection Models

Nine portfolio-selection policies are compared: feasible random selection, priority ranking, priority-to-cost greedy selection, dependency-aware greedy selection, linear MILP, hard- and soft-dependency MILP, and QUBO formulations with and without dependencies. The first four methods serve as heuristic baselines, while the MILP formulations provide linear, hard-dependency, and soft-dependency references. Both QUBO models are solved using classical simulated annealing [Kirkpatrick et al. 1983].

For n candidate assets, let $x_i \in \{0, 1\}$ indicate whether asset i is selected, r_i its predicted priority, c_i its migration cost, and B the available budget. The linear MILP reference follows the 0/1 knapsack formulation [Martello and Toth 1990]:

$$\max_{x\in\{0,1\}^n} \sum_{i=1}^{n} r_i x_i \quad \text{s.t.} \quad \sum_{i=1}^{n} c_i x_i \leq B. \tag{1}$$

Mandatory prerequisites are represented by $x_i \leq x_j, \qquad \forall (i, j) \in D_{\text{hard}}.$

The common soft objective combines selected priority, weighted mandatory violations, and optional coordination benefits:

$$U(x) = \sum_i r_i x_i - \lambda_D \sum_{(i,j)\in D_{\text{hard}}} d_{ij} x_i (1 - x_j) + \lambda_G \sum_{(i,j)\in D_{\text{soft}}} d_{ij} x_i x_j. \tag{2}$$

The QUBO formulation incorporates the budget condition through a binary-encoded nonnegative slack variable s. The slack is encoded as $s = \sum_{k=0}^{m-1} 2^k s_k$, with $s_k \in \{0, 1\}$ and $m = \lceil \log_2(\tilde{B}+1) \rceil$. Costs are discretized in units of 100 as $\tilde{c}_i = \max\{1, \text{round}(c_i/100)\}$ and $\tilde{B} = \max\{1, \text{round}(B/100)\}$, whereas MILP retains the original monetary values. Therefore, the two formulations approximate the same portfolio-selection problem, although cost discretization makes their feasible representations not algebraically identical. Following standard QUBO constructions [Glover et al. 2019, Lucas 2014], the dependency-free model is

$$Q_{\text{base}}(x, s) = -\sum_i r_i x_i + \lambda_B \left(\sum_i \tilde{c}_i x_i + s - \tilde{B} \right)^2. \tag{3}$$

where $\tilde{c}_i$ and $\tilde{B}$ denote discretized costs and budget. The dependency-aware extension is

$$Q_{\text{dep}}(x, s) = Q_{\text{base}}(x, s) + \lambda_D \sum_{(i,j)\in D_{\text{hard}}} d_{ij} x_i (1 - x_j) - \lambda_G \sum_{(i,j)\in D_{\text{soft}}} d_{ij} x_i x_j. \tag{4}$$

Independent calibration evaluates $\lambda_B \in \{0.20, 0.50, 1.00, 2.00\}$ and $\lambda_D \in \{0.20, 0.50, 1.00\}$, with $\lambda_G = 0.03$. Selection prioritizes budget feasibility and dependency reduction while preserving at least 95% of the dependency-free priority. The resulting values, $\lambda_B = 1.0$ and $\lambda_D = 0.5$, remain fixed during evaluation.

A mandatory dependency (i, j) is violated when $x_i = 1$ and $x_j = 0$. Technical coherence is defined over mandatory edges whose source asset is selected:

$$C(x) = 1 - \frac{\sum_{(i,j)\in D_{\text{hard}}} I[x_i = 1 \wedge x_j = 0]}{\sum_{(i,j)\in D_{\text{hard}}} I[x_i = 1]}. \tag{5}$$

When no selected source asset has a relevant mandatory dependency, coherence is treated as not applicable. Differences among policies are assessed through paired bootstrap comparisons over the 135 common scenarios. The priority gap reported in Table 1 is computed scenario-wise from selected predicted priority relative to the linear MILP reference and then averaged over the 135 common scenarios.

3. Results and Discussion

3.1. Priority Estimation

Logistic Regression showed the most consistent performance across the three evaluation inventories and was selected to generate the optimization benefits. On the system-grouped test partitions, AUC-ROC ranged from 0.928 to 0.952, average precision from 0.854 to 0.881, accuracy from 0.843 to 0.883, F_1 from 0.732 to 0.809, and Brier score from 0.082 to 0.108. Sigmoid and isotonic calibration did not consistently improve the Brier score;

therefore, the original probabilities were retained. Grouping the partitions by organizational system reduces optimistic estimates caused by related assets appearing in both training and test data. Nevertheless, these results measure agreement with the simulator-defined priority construct rather than validity against empirical organizational decisions.

3.2. Optimization Performance

Table 1 presents the aggregated performance of the nine portfolio-selection policies over the 135 common scenarios.

Table 1. Aggregated performance by selection policy.

Method	Pred. priority	Viol.	Coherence	Priority gap (%)	Runtime (s)
Random feasible	9.557	8.044	0.199	71.669	0.0002
Priority ranking	24.988	4.519	0.507	25.197	< 0.0001
Priority/cost greedy	32.704	10.904	0.346	0.206	0.0001
Dependency-aware greedy	27.842	0.000	1.000	15.646	0.0051
Linear MILP	32.760	10.652	0.350	0.000	0.0148
Hard-dependency MILP	30.151	0.000	1.000	8.566	0.0244
Soft-dependency MILP	31.636	3.030	0.721	3.674	0.0448
QUBO SA without dependencies	32.658	10.963	0.344	0.322	16.4030
QUBO SA with dependencies	31.890	7.859	0.479	2.487	16.2822

The comparison reveals a trade-off between selected predicted priority and dependency consistency. Cost-aware policies closely approached the linear optimum, whereas priority ranking performed substantially worse, showing that priority alone is insufficient under a budget constraint. Dependency-blind methods also produced more unmet prerequisites. Hard-dependency policies eliminated mandatory violations at the cost of lower selected priority. Soft-dependency MILP provided the most favorable compromise under the adopted objective, preserving most of the linear-MILP benefit while substantially reducing violations and improving coherence. This result supports soft dependency treatment in staged migration settings, where temporary inconsistencies may be tolerated but should remain penalized.

3.3. Effect of Dependency Modeling

To isolate the contribution of dependency terms, Table 2 reports the paired differences between the two QUBO formulations. Differences are computed as dependency-aware QUBO minus dependency-free QUBO over the same 135 scenarios.

Table 2. Paired ablation between the QUBO formulations.

Metric	Mean difference	95% CI
Selected priority	-0.768	$[-0.986, -0.563]$
Mandatory violations	-3.104	$[-3.867, -2.281]$
Weighted violations	-2.149	$[-2.722, -1.596]$
Common soft objective	$+0.306$	$[0.221, 0.397]$

Dependency-aware QUBO reduced mandatory and weighted violations and improved the common soft objective, while decreasing selected predicted priority by 0.768 on average. The paired confidence intervals exclude zero, indicating consistent differences in portfolio structure. Compared with dependency-aware QUBO, soft-dependency

MILP produced fewer mandatory violations, greater technical coherence, and a mean advantage of 1.418 in the common soft objective, although QUBO retained slightly more selected predicted priority. Heuristic and MILP policies completed in milliseconds, whereas the two QUBO variants required approximately 16.3 s per scenario because of repeated multistart simulated annealing. Therefore, the results provide no evidence of computational advantage for QUBO at the evaluated scales. Hard dependencies are preferable when violations are unacceptable, whereas soft formulations better support staged migration.

4. Conclusion

This study formulated PQC migration prioritization as a budget-constrained portfolio-selection problem combining predicted priority, migration cost, and technical dependencies. System-grouped priority estimates remained consistent across the three synthetic inventories. In the optimization stage, dependency-blind methods, including linear MILP, priority-to-cost greedy selection, and dependency-free QUBO, achieved the highest selected priority but produced less coherent portfolios.

Adding dependencies to QUBO reduced mean mandatory violations from 10.963 to 7.859 and increased coherence from 0.344 to 0.479, with a limited decrease in selected priority. Under the adopted soft objective and penalty weights, soft-dependency MILP achieved the best trade-off and substantially lower runtime. These results support the research hypothesis that explicit dependency modeling improves migration coherence at a moderate priority cost.

Overall, QUBO provided a flexible quadratic representation of budget and dependency interactions, but did not outperform the classical methods in runtime or overall solution quality at the evaluated scales. Limitations include synthetic inventories, simulator-defined priorities and costs, simplified pairwise dependencies, and classical SA. Future work should evaluate empirical inventories and richer dependency structures.

References

CISA, NSA, and NIST (2023). Quantum-readiness: Migration to post-quantum cryptography. Cybersecurity and Infrastructure Security Agency, National Security Agency, and National Institute of Standards and Technology. Available at: `https://www.cisa.gov/resources-tools/resources/quantum-readiness-migration-post-quantum-cryptography`. Accessed: 16 Jun. 2026.

ETSI (2020). ETSI TR 103 619 V1.1.1: CYBER; migration strategies and recommendations to quantum safe schemes. Technical report, European Telecommunications Standards Institute, Sophia Antipolis. Available at: `https://www.etsi.org/deliver/etsi_tr/103600_103699/103619/01.01.01_60/tr_103619v010101p.pdf`. Accessed: 16 Jun. 2026.

Figueira, L. d. O., Duarte, L. O., and Silva, R. L. (2026). PQC migration prioritization with machine learning and QUBO. GitHub repository. Version 1.0.0. Replication package containing the source code, notebook, synthetic datasets, experimental outputs, and replication instructions. Available at: `https://github.com/larissaoliveira-lgtm/pqc-migration-prioritization-ml-qubo`. Accessed: 5 Aug. 2026.

Glover, F., Kochenberger, G., and Du, Y. (2019). Quantum bridge analytics I: A tutorial on formulating and using QUBO models. *4OR*, 17:335–371.

Hasan, K. F., Simpson, L., Rezazadeh Baee, M. A., Islam, C., Rahman, Z., Armstrong, W., Gauravaram, P., and McKague, M. (2024). A framework for migrating to post-quantum cryptography: Security dependency analysis and case studies. *IEEE Access*, 12:23427–23450.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Lucas, A. (2014). Ising formulations of many NP problems. *Frontiers in Physics*, 2:5.

Martello, S. and Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Chichester.

NIST (2024a). FIPS 203: Module-lattice-based key-encapsulation mechanism standard. Federal Information Processing Standards Publication 203, National Institute of Standards and Technology, Gaithersburg, MD.

NIST (2024b). FIPS 204: Module-lattice-based digital signature standard. Federal Information Processing Standards Publication 204, National Institute of Standards and Technology, Gaithersburg, MD.

NIST (2024c). FIPS 205: Stateless hash-based digital signature standard. Federal Information Processing Standards Publication 205, National Institute of Standards and Technology, Gaithersburg, MD.

Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Los Alamitos, CA. IEEE Computer Society Press.

Silva, R. and Duarte, L. (2026a). Engineering trust in LLM supply chains through hybrid post-quantum artifact signatures. In *Proceedings of the 20th IEEE International Workshop on Security, Trust, and Privacy for Software Applications (STPSA 2026), held in conjunction with the 50th IEEE Computer Society Annual International Conference on Computers, Software, and Applications (COMPSAC 2026)*, Madrid, Spain. IEEE. Workshop paper listed in the official COMPSAC 2026 program.

Silva, R. and Duarte, L. (2026b). Securing LLM software supply chains: A layered lifecycle framework with hybrid post-quantum artifact signing. In *Proceedings of IEEE IDS 2026*. IEEE. Camera-ready manuscript.

Silva, R. L. (2026). Toward quantum-resilient software supply chains: A DevSecOps case study with hybrid post-quantum artifact signing. In *Proceedings of the 2026 IEEE International Conference on Quantum Communications, Networking, and Computing (QCNC)*, Kobe, Japan. IEEE. IEEE Xplore document 11500303. Available at: `https://ieeexplore.ieee.org/document/11500303`.