Weight Setting Methods (WSMs) for QUBO Penalty Weights

UB, MQC, VLM, MOMC, MOC and Recent Extensions

Jehn-Ruey Jiang

Department of Computer Science and Information Engineering
National Central University, Taiwan

Primary sources: Jiang and Chu, IEEE Access, 2023; Jiang, Shu, and Lin, IEEE Access, 2024.

Why Do We Need Penalty-Weight Setting?

For a constrained binary optimization problem, write the QUBO as

$$f(\mathbf{x}) = \alpha g(\mathbf{x}) + c(\mathbf{x}),$$

where $g(\mathbf{x}) = 0$ for feasible solutions and $g(\mathbf{x}) > 0$ for infeasible solutions.

- α too small: infeasible states may have lower energy than feasible states.
- α too large: the cost landscape can be compressed relative to the penalty scale, potentially degrading search performance and hardware precision utilization.
- Therefore, we want a penalty that is **large enough for feasibility, but not unnecessarily large**.

Jiang and Chu, "Classifying and Benchmarking Quantum Annealing Algorithms Based on QUBO for Solving NP-Hard Problems," IEEE Access, vol. 11, pp. 104165–104180, 2023, Sec. II-C.

A Sufficient Penalty Condition

Let $\mathbf{y}$ be an optimal feasible solution and S the set of infeasible solutions. To prevent an infeasible $\mathbf{x} \in S$ from beating $\mathbf{y}$,

$$c(\mathbf{y}) < \alpha g(\mathbf{x}) + c(\mathbf{x}), \qquad \forall \mathbf{x} \in S.$$

Hence a valid penalty weight should satisfy

$$\boxed{\alpha > \max_{\mathbf{x} \in S} \frac{c(\mathbf{y}) - c(\mathbf{x})}{g(\mathbf{x})}}.$$

The difficulty is that the right-hand side depends on information that is generally unavailable before solving the problem. WSMs construct computable estimates or bounds.

Jiang and Chu, IEEE Access, 2023, Eqs. (5)–(7); Jiang, Shu, and Lin, IEEE Access, 2024, Eqs. (5)–(7).

Five WSMs Used in the Two Papers

WSM	Name	Main idea
UB	Upper Bound	Use an upper bound on the objective magnitude.
MQC	Maximum QUBO Coefficient	Use the largest coefficient in the cost QUBO.
VLM	Verma–Lewis Method	Estimate the maximum objective change caused by a one-bit flip.
MOMC	Maximum Objective / Minimum Constraint	Normalize the VLM objective-change estimate by the smallest positive constraint change.
MOC	Maximum Objective / Corresponding Constraint	Use the ratio of objective change to the corresponding constraint change, bit by bit.

Jiang and Chu, IEEE Access, 2023, Table 1; Jiang, Shu, and Lin, IEEE Access, 2024, Table 1. Original WSM sources: Verma and Lewis, Discrete Optimization, 2022; Ayodele, EvoCOP 2022.

UB and MQC: Simple Scale-Based Rules

Let C be the QUBO matrix of the cost function $c(\mathbf{x})$, and let $\mathbf{z}$ be the all-ones vector.

Upper Bound (UB)

$$\alpha_{\mathrm{UB}} = \mathbf{z}^T C \mathbf{z}, \qquad z_i = 1.$$

It represents an objective upper bound under the positive-coefficient assumption used in the papers.

Maximum QUBO Coefficient (MQC)

$$\alpha_{\mathrm{MQC}} = \max_i \max_j C_{ij}.$$

Very inexpensive, but it uses only the largest individual coefficient rather than aggregate bit-flip effects.

Jiang and Chu, IEEE Access, 2023, Table 1 and Sec. II-C; Jiang, Shu, and Lin, IEEE Access, 2024, Table 1 and Sec. II-A.

VLM: Estimate the Worst Objective Change

For each bit i, define an objective-change bound

$$W_i^c = \max\left(-C_{ii} - \sum_{j=1}^{n} \min(C_{ij}, 0),\ C_{ii} + \sum_{j=1}^{n} \max(C_{ij}, 0)\right).$$

Then

$$\boxed{\alpha_{\mathrm{VLM}} = \max_i W_i^c.}$$

- Interpretable as a conservative estimate of how much the objective can improve through a bit change.
- It estimates the numerator $c(\mathbf{y}) - c(\mathbf{x})$ in the sufficient-condition ratio.
- It does **not** explicitly account for the magnitude of the constraint violation in the denominator.

Verma and Lewis, "Penalty and partitioning techniques to improve performance of QUBO solvers," Discrete Optimization, vol. 44, 100594, 2022; Jiang and Chu, IEEE Access, 2023, Table 1.

MOMC: Include the Minimum Constraint Change

Let G denote the QUBO matrix of $g(\mathbf{x})$. Define the bit-wise constraint-change quantity

$$W_i^g = \min\left(-G_{ii} - \sum_{j \neq i} \min(G_{ij}, 0),\ G_{ii} + \sum_{j \neq i} \max(G_{ij}, 0)\right),$$

and the smallest positive change

$$\gamma = \min_{i : W_i^g > 0} W_i^g.$$

MOMC sets

$$\boxed{\alpha_{\mathrm{MOMC}} = \max\left(1, \frac{\alpha_{\mathrm{VLM}}}{\gamma}\right).}$$

Compared with VLM, MOMC explicitly incorporates a lower estimate of positive constraint change.

Ayodele, "Penalty Weights in QUBO Formulations: Permutation Problems," EvoCOP 2022, LNCS, pp. 159–174; Jiang and Chu, IEEE Access, 2023, Table 1.

MOC: Match Objective and Constraint Changes

MOC uses the objective and constraint changes associated with the **same bit**:

$$\boxed{\alpha_{\mathrm{MOC}} = \max\left(1, \max_{i:W_i^g>0} \left|\frac{W_i^c}{W_i^g}\right|\right).}$$

- VLM: objective-change information only.
- MOMC: worst objective change divided by the globally smallest positive constraint change.
- MOC: pairs each objective change with its corresponding constraint change before taking the maximum ratio.

Thus MOC more directly mirrors the ratio appearing in the sufficient penalty condition.

Ayodele, EvoCOP 2022; Jiang and Chu, IEEE Access, 2023, Table 1 and Sec. II-C; Jiang, Shu, and Lin, IEEE Access, 2024, Table 1.

Conceptual Comparison of the Five WSMs

WSM	Uses objective scale	Uses constraint change	Character
UB	Yes, global bound	No	Simple and potentially conservative
MQC	Yes, single coefficient	No	Cheapest scale heuristic
VLM	Yes, bit-flip change	No	Better reflects local objective variation
MOMC	Yes	Yes, minimum positive	Conservative ratio-based refinement
MOC	Yes	Yes, corresponding bit	More tightly couples objective and constraint effects

Key progression: coefficient scale $\rightarrow$ objective change $\rightarrow$ objective/constraint change ratio.

Synthesis of Jiang and Chu, IEEE Access, 2023, Sec. II-C and Table 1; Jiang, Shu, and Lin, IEEE Access, 2024, Sec. II-A and Table 1.

Supplement: Exact and Sequential Penalty Weights

Garcia, Ayodele, and Moraglio proposed a general approach that goes beyond fixed coefficient heuristics.

- **Exact penalty bound:** derive a computable upper bound on the smallest penalty that preserves the feasible global optimum.
- **Sequential penalty:** iteratively tighten/improve the penalty through repeated solver calls.
- The approach was evaluated on minimum cut, TSP, and multidimensional knapsack problems using a Digital Annealer.

Interpretation

Static WSMs estimate a safe α from QUBO structure; sequential methods spend additional optimization effort to reduce unnecessary over-penalization.

M. Diez Garcia, M. Ayodele, and A. Moraglio, "Exact and Sequential Penalty Weights in Quadratic Unconstrained Binary Optimisation with a Digital Annealer," GECCO '22 Companion, 2022, doi:10.1145/3520304.3528925.

Supplement: Multiple Penalty Weights

A single common α may be inefficient when different constraints have very different sensitivities.

$$F(\mathbf{x}) = c(\mathbf{x}) + \sum_{k=1}^{K} \alpha_k g_k(\mathbf{x}).$$

Recent work extends automatic penalty setting to **multiple constraint-specific weights**:

- first determine a common penalty using exact/sequential search;
- analyze constraint sensitivity;
- adapt individual α_k values instead of forcing all constraints to share one weight.

This direction is especially relevant to QUBOs containing heterogeneous constraint families.

J. Liu and A. Moraglio, "A Framework for Automatically Setting Multiple Penalty Weights in Quadratic Unconstrained Binary Optimization," GECCO '25, pp. 575–578, 2025, doi:10.1145/3712255.3726759.

Supplement: Solver-in-the-Loop / Adaptive Tuning

A broader modern direction is to treat penalty setting as an **outer-loop optimization problem** rather than a one-shot formula.

1. Choose initial penalty weight(s).
2. Build and solve the QUBO.
3. Measure feasibility, violation magnitude, and/or solution quality.
4. Increase weights for persistently violated constraints; optionally relax already-satisfied constraints.
5. Repeat until a stopping condition is reached.

This naturally connects WSMs to Bayesian optimization, feedback control, or other hyperparameter-search methods.

Important distinction

UB/MQC/VLM/MOMC/MOC are primarily **formula-based static WSMs**; adaptive methods use **solver feedback** and therefore require multiple QUBO evaluations.

For the static taxonomy: Jiang et al., IEEE Access, 2024. For a recent feedback-style example: automatic QUBO penalty-weight tuning in Appl. Sci., vol. 16, 1690, 2026.

Takeaways

- The theoretical target is to make every infeasible state energetically worse than the optimal feasible state.
- **UB/MQC** use coarse objective scales.
- **VLM** estimates objective changes caused by bit flips.
- **MOMC/MOC** additionally incorporate constraint changes and more closely approximate the required penalty ratio.
- Newer approaches move toward **exact/sequential**, **multiple-weight**, and **adaptive solver-in-the-loop** penalty setting.

Penalty-weight setting is part of the optimization problem itself.

Primary synthesis: Jiang and Chu, IEEE Access, 2023; Jiang, Shu, and Lin, IEEE Access, 2024.

References

- J.-R. Jiang and C.-W. Chu, "Classifying and Benchmarking Quantum Annealing Algorithms Based on Quadratic Unconstrained Binary Optimization for Solving NP-Hard Problems," *IEEE Access*, vol. 11, pp. 104165–104180, 2023, doi:10.1109/ACCESS.2023.3318206.
- J.-R. Jiang, Y.-C. Shu, and Q.-Y. Lin, "Benchmarks and Recommendations for Quantum, Digital, and GPU Annealers in Combinatorial Optimization," *IEEE Access*, vol. 12, pp. 125014–125031, 2024, doi:10.1109/ACCESS.2024.3455436.
- A. Verma and M. Lewis, "Penalty and partitioning techniques to improve performance of QUBO solvers," *Discrete Optimization*, vol. 44, 100594, 2022, doi:10.1016/j.disopt.2020.100594.
- M. Ayodele, "Penalty Weights in QUBO Formulations: Permutation Problems," in *EvoCOP 2022*, LNCS, pp. 159–174, 2022, doi:10.1007/978-3-031-04148-8_11.
- M. Diez Garcia, M. Ayodele, and A. Moraglio, "Exact and Sequential Penalty Weights in Quadratic Unconstrained Binary Optimisation with a Digital Annealer," *GECCO '22 Companion*, 2022, doi:10.1145/3520304.3528925.
- J. Liu and A. Moraglio, "A Framework for Automatically Setting Multiple Penalty Weights in Quadratic Unconstrained Binary Optimization," *GECCO '25*, pp. 575–578, 2025, doi:10.1145/3712255.3726759.