Don't Waste the Population: Post-Anneal Refinement for Combinatorial Optimization

Yuma Ichikawa
Fujitsu Limited
RIKEN Center for AIP

Abstract

Parallel annealers produce many candidate solutions, but selecting and polishing only the best rounded replica leaves the rest of the population unused. We study this post-processing choice in Parallel Quasi-Quantum Annealing (PQQA) and introduce Iterative Path-Relinking Polish (I-PRP), a deterministic refinement that leaves the annealing loop unchanged. I-PRP polishes a small set of elites, follows cost-guided paths from the best polished elite toward the others, and polishes the intermediate states to search for better basins. A newly discovered solution enters the population only under strict improvement. For a fixed rounded population, the method cannot worsen the baseline polish, terminates after finitely many updates, and returns the baseline solution when all selected elites share one basin. Its terminal certificate covers every polished intermediate on the paths induced by the final elite population. QUBO and graph-coloring benchmarks show improvements on RBtest MaxClique, many G-set MaxCut instances, and selected random regular graphs. A paired study of 90 saved populations finds 29 strict improvements and no regressions; matched-runtime controls isolate the benefit of refinement from additional annealing.

1 Introduction

Combinatorial optimization underlies scheduling, routing, and graph analysis, but many of its basic formulations are computationally hard [Papadimitriou and Steiglitz, 1998, Korte and Vygen, 2012, Karp, 1972]. Practical solvers must balance solution quality with the computational cost of each new instance. Learning-based methods can amortize heuristic design across related instances [Bengio et al., 2021, Cappart et al., 2023, Li et al., 2018, Dai et al., 2017, Karalias and Loukas, 2020], while training-free methods optimize each instance directly through sampling, relaxation, or local search [Kirkpatrick et al., 1983, Lourenço et al., 2003]. We focus on improving the discrete solution returned by an existing training-free pipeline.

Recent solvers expose more of their search to parallel hardware. Locally informed samplers use energy differences to guide discrete proposals [Grathwohl et al., 2021, Zhang et al., 2022, Sun et al., 2023a,b, Goshvadi et al., 2023]. Continuous-relaxation annealers optimize soft variables before driving them toward discrete configurations [Ichikawa, 2024]. Parallel Quasi-Quantum Annealing (PQQA) combines this relaxation with a population of interacting replicas [Ichikawa and Arai, 2025]. The resulting candidates can occupy different regions of the objective landscape.

The post-processing pipeline considered here retains only the best rounded PQQA replica and applies a greedy one-flip polish. This selection uses the population to choose a starting point, but does not exploit relationships among the discarded candidates. Two rounded replicas can descend to different local optima even when their objective values are similar. Their coordinate disagreements then specify a route between those optima, along which further local searches may discover a better solution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Path relinking explores such routes by gradually introducing the attributes of one reference solution into another [Glover et al., 2000, Ribeiro and Resende, 2012]. It has been combined with elite archives in greedy randomized adaptive search procedures (GRASP), tabu search, and quadratic optimization [Festa et al., 2002, Martí et al., 2009, Wang et al., 2012b]. Here the reference solutions are already available in the rounded annealing population. The question is how to use them as a deterministic refinement after the upstream search has finished.

We introduce **Iterative Path-Relinking Polish (I-PRP)**. A pass polishes the top rounded elites, constructs cost-guided paths from the best polished elite to the others, and polishes each intermediate state. These paths may move uphill; the returned incumbent changes only when a polished candidate improves its objective. A successful pass inserts that candidate into the elite tuple and repeats the search. We prove non-degradation relative to the baseline on the same rounded population and finite termination of the exact iteration. At termination, none of the polished states on the final population's paths improves the incumbent. This certificate describes the routes actually explored and does not require the population to cover all local moves.

The experiments evaluate this refinement on maximum independent set, maximum clique, maximum cut, and graph coloring. The reported gains are most pronounced on RBtest MaxClique and many large G-set MaxCut instances; categorical relinking also resolves the residual coloring conflicts on `anna` and `jean`. Paired controls on saved populations separate endpoint selection, intermediate re-polishing, and iteration. Comparisons with longer and restarted PQQA runs then measure the value of this refinement under matched time budgets. Together, these results motivate retaining several basin representatives after annealing, even when only one solution is ultimately returned.

2 Background

For a positive integer m, write $[m] = \{1, \dots, m\}$. Binary configurations lie in $\{0,1\}^N$, where N is a positive integer, and $H(\boldsymbol{x}, \boldsymbol{y}) = |\{i \in [N] : x_i \neq y_i\}|$ denotes Hamming distance. All objective values are finite real numbers, and smaller values are better unless an experimental score is explicitly identified as a maximization objective.

Discrete objectives and penalties. For an instance C with original cost f_0 and M constraints, a penalty formulation is [Lucas, 2014, Kochenberger et al., 2014]

$$\min_{\boldsymbol{x}\in\{0,1\}^N} l(\boldsymbol{x}; C, \boldsymbol{\lambda}), \qquad l(\boldsymbol{x}; C, \boldsymbol{\lambda}) = f_0(\boldsymbol{x}; C) + \sum_{i=1}^{M} \lambda_i v_i(\boldsymbol{x}; C), \tag{1}$$

where $\lambda_i > 0$, $v_i \geq 0$, and all penalties vanish exactly on feasible configurations. When interpreting global penalized optima as constrained optima, we assume that the feasible set is nonempty and every infeasible configuration has penalized cost strictly larger than the best feasible cost. The refinement guarantees below do not need this exact-penalty assumption: they hold for the fixed scalar objective $f(\boldsymbol{x}) = l(\boldsymbol{x}; C, \boldsymbol{\lambda})$ itself. In particular, descent of a general penalty objective alone is not a feasibility guarantee.

A quadratic unconstrained binary optimization (QUBO) objective has the form $f(\boldsymbol{x}) = \boldsymbol{x}^\top Q\boldsymbol{x} + \boldsymbol{q}^\top \boldsymbol{x}$, with $Q = Q^\top \in \mathbb{R}^{N\times N}$ and $\boldsymbol{q} \in \mathbb{R}^N$. Under $s_i = 2x_i - 1$ for $i \in [N]$, it can be written, up to an additive constant, as the Ising objective $f(\boldsymbol{s}) = -\boldsymbol{s}^\top J\boldsymbol{s}/2 - \boldsymbol{h}^\top \boldsymbol{s}$, with $J = J^\top$ and $J_{ii} = 0$. These representations share the same discrete search structure.

Graph objectives. The binary benchmarks differ in their discrete scores and penalty terms. For a simple graph $G = ([N], E)$, an independent set selects vertices with no selected edge. A standard QUBO encoding is

$$f_{\mathrm{MIS}}(\boldsymbol{x}) = -\sum_{i=1}^{N} x_i + \lambda \sum_{\{i,j\}\in E} x_i x_j, \qquad \lambda > 1. \tag{2}$$

The first term rewards cardinality and the second penalizes selected edges. With this strict penalty condition, every one-flip local optimum is feasible: if a selected vertex has $d \geq 1$ selected neighbors, removing it changes the objective by $1 - \lambda d < 0$. On a feasible state, minimizing this objective is equivalent to maximizing the independent-set size. Maximum clique has the same encoding with

the edge set replaced by the graph's nonedges. This gives the same feasibility argument for clique solutions. The strict condition matters; at $\lambda = 1$, a violating move can be neutral and the strict polish need not remove it.

For a weighted graph, a cut assigns the two sides by $x_i \in \{0, 1\}$. If edge $\{i, j\}$ has real weight w_{ij}, its cut value and minimization objective are

$$c(\boldsymbol{x}) = \sum_{\{i,j\} \in E} w_{ij}(x_i + x_j - 2x_i x_j), \qquad f_{\mathrm{cut}}(\boldsymbol{x}) = -c(\boldsymbol{x}). \tag{3}$$

Thus a lower optimization objective corresponds to a larger reported cut. These encodings explain the relation between local descent and the benchmark scores. The general I-PRP analysis depends only on the resulting fixed objective, so it also applies to other penalty choices; the feasibility conclusion above specifically requires the stated encoding and $\lambda > 1$.

Parallel quasi-quantum annealing. PQQA maintains $S \geq 1$ relaxed replicas $\boldsymbol{p}^{(s)} \in [0, 1]^N$ and applies gradient-based sampling [Ichikawa and Arai, 2025]. Its per-replica relaxed objective is

$$\hat{r}(\boldsymbol{p}; C, \boldsymbol{\lambda}, \gamma) = \hat{l}(\boldsymbol{p}; C, \boldsymbol{\lambda}) + \gamma s(\boldsymbol{p}), \qquad s(\boldsymbol{p}) = \sum_{i=1}^{N} \big[1 - (2p_i - 1)^4\big], \tag{4}$$

where $\hat{l}$ is differentiable and agrees with l at binary vertices. The concave regularizer s is largest at the half-integral vector and vanishes at binary vertices. Annealing γ from negative to positive first favors interior values and then penalizes them [Ichikawa, 2024]. A cross-replica term encourages population diversity.

The final replicas are rounded by $\widetilde{x}_i^{(s)} = 1$ if $p_i^{(s)} \geq 1/2$ and 0 otherwise. Write the rounded population as $\widetilde{\mathcal{P}} = (\widetilde{\boldsymbol{x}}^{(1)}, \ldots, \widetilde{\boldsymbol{x}}^{(S)})$. We order states by increasing objective, breaking ties lexicographically with $0 < 1$. The baseline post-processing in this paper polishes only the first state in this order. I-PRP uses the same rounded population and changes this final refinement step.

One-flip polish and basins. Let $\sigma_i \boldsymbol{x}$ denote $\boldsymbol{x}$ with coordinate i flipped, and define $\Delta_i(\boldsymbol{x}) = f(\sigma_i \boldsymbol{x}) - f(\boldsymbol{x})$. For the symmetric QUBO,

$$\Delta_i^{\mathrm{QUBO}}(\boldsymbol{x}) = (1 - 2x_i)\big[q_i + Q_{ii} + 2(Q\boldsymbol{x})_i - 2Q_{ii}x_i\big]. \tag{5}$$

The corresponding Ising difference is $\Delta_i^{\mathrm{spin}}(\boldsymbol{s}) = 2s_i[(J\boldsymbol{s})_i + h_i]$. Both can be computed from a cached matrix-vector product; Appendix C.1 derives the differences and cache updates.

The deterministic map $\mathsf{Polish} : \{0, 1\}^N \to \{0, 1\}^N$ repeatedly performs

$$\boldsymbol{x} \leftarrow \sigma_{i^*} \boldsymbol{x}, \qquad i^* = \min \arg\min_{j \in [N]} \Delta_j(\boldsymbol{x}), \qquad \text{while } \min_{j \in [N]} \Delta_j(\boldsymbol{x}) < 0. \tag{6}$$

The smallest-index rule resolves ties. Each accepted move strictly decreases f, so no state is revisited and the finite search terminates. Its output is one-flip locally optimal, meaning that every one-flip difference is nonnegative. Consequently, Polish is idempotent. For each locally optimal state $\boldsymbol{x}^\circ$, define its basin as

$$\mathcal{B}(\boldsymbol{x}^\circ) = \{\boldsymbol{x} \in \{0, 1\}^N : \mathsf{Polish}(\boldsymbol{x}) = \boldsymbol{x}^\circ\}.$$

Determinism makes these basins a partition of the cube, including landscapes with tied objective values. They describe the outcome of this particular polish, rather than an assumed geometric neighborhood. The baseline selects one basin; the remaining replicas can provide representatives of others.

3 Method

I-PRP uses coordinate disagreements between polished elites to construct search paths (Figure 1). The path may increase the objective temporarily, while a separate incumbent retains the best polished solution found. All formal results use complete paths, exact objective comparisons, and the deterministic polish defined above.

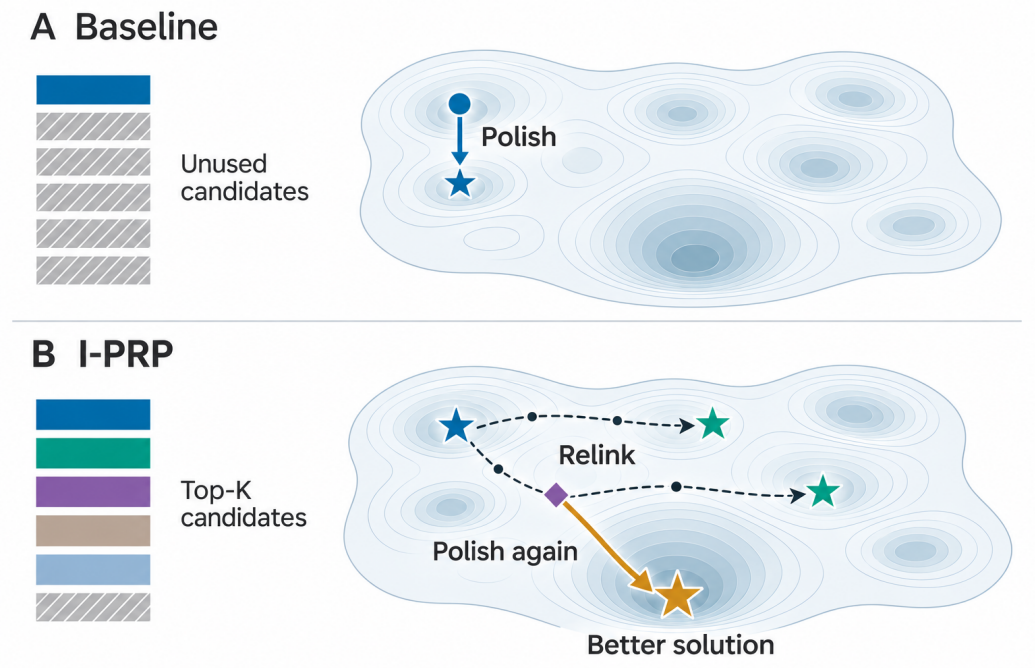

Figure 1: Post-anneal refinement. (A) The baseline polishes only the best rounded replica. (B) I-PRP polishes the top-K elites and follows separate paths from the best polished elite to the others. Re-polishing an intermediate state can reach a better basin. The paths retain their endpoints throughout each pass; the landscape is schematic.

For two states $\boldsymbol{x}_a, \boldsymbol{x}_b \in \{0,1\}^N$, path relinking starts at $\boldsymbol{c}^{(0)} = \boldsymbol{x}_a$. At step t, let $D_t = \{i : c_i^{(t)} \neq x_{b,i}\}$. If $D_t \neq \emptyset$, choose the lowest-cost flip toward the target:

$$i_t^* = \min \arg\min_{j \in D_t} \Delta_j(\boldsymbol{c}^{(t)}), \qquad \boldsymbol{c}^{(t+1)} = \sigma_{i_t^*} \boldsymbol{c}^{(t)}. \tag{7}$$

Each move removes one coordinate from the disagreement set, so the path reaches $\boldsymbol{x}_b$ after exactly $H(\boldsymbol{x}_a, \boldsymbol{x}_b)$ steps. Unlike local descent, the move need not improve f. This permits the path to leave a local optimum before a fresh polish is applied [Glover et al., 2000, Ribeiro and Resende, 2012].

3.1 One pass: Path-Relinking Polish

Sort the rounded population by objective and lexicographic order, and take its top-K prefix

$$\mathcal{P} = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}), \qquad 1 \leq K \leq S. \tag{8}$$

Repeated states are allowed. The one-pass map $\Phi_K : (\{0,1\}^N)^K \to \{0,1\}^N$ processes this tuple as follows.

Stage 1: polish the elites. Compute $\mathsf{Polish}(\boldsymbol{x}^{(k)})$ for all $k \in [K]$, and sort these outputs by objective and lexicographic order. Denote the sorted polished states by $\boldsymbol{x}^{(1)\star}, \ldots, \boldsymbol{x}^{(K)\star}$. Their indices refer to this new order, not to the raw input ranks. Initialize the incumbent to $\boldsymbol{x}^\star = \boldsymbol{x}^{(1)\star}$ and write $b(\mathcal{P}) = f(\boldsymbol{x}^{(1)\star})$. These states serve as basin representatives. Raw objective order need not agree with polished objective order, so this step alone can improve on polishing only the best rounded replica.

Stage 2: relink the polished endpoints. For $k = 2, \ldots, K$ in that order, apply (7) from $\boldsymbol{x}^{(1)\star}$ to $\boldsymbol{x}^{(k)\star}$. Write $\boldsymbol{c}^{(t,k)}$ for its state after t moves, $D_t^{(k)}$ for its disagreement set, and $d_k = |D_0^{(k)}|$. The endpoints remain fixed throughout the pass, even if another candidate improves the incumbent. Duplicate endpoints have $d_k = 0$ and require no path moves.

Stage 3: polish the path states. After each path move, polish a copy of the visited state:

$$\boldsymbol{z}^{(t,k)} = \mathsf{Polish}(\boldsymbol{c}^{(t,k)}), \qquad 1 \le t \le d_k. \tag{9}$$

Replace $\boldsymbol{x}^\star$ by $\boldsymbol{z}^{(t,k)}$ only when $f(\boldsymbol{z}^{(t,k)}) < f(\boldsymbol{x}^\star)$. Polishing does not alter the path state or its target. Return $\Phi_K(\mathcal{P}) = \boldsymbol{x}^\star$ after all paths have been processed. Strict promotion retains the earlier incumbent on ties, so the pass is deterministic.

Equivalently, the returned value is the minimum over the best polished elite and all polished path states:

$$f(\Phi_K(\mathcal{P})) = \min\left(\{b(\mathcal{P})\} \cup \left\{f\big(\mathsf{Polish}(\boldsymbol{c}^{(t,k)})\big) : 2 \le k \le K,\ 1 \le t \le d_k\right\}\right). \tag{10}$$

This identity separates the gain from polishing more elites from the further gain obtained by relinking. Its incumbent invariant is proved in Appendix B.1.

Re-polishing lets the search change coordinates outside the endpoint disagreement set. The path therefore supplies starting points for descents into basins that need not contain either endpoint. Appendix A.1 illustrates this mechanism with a three-variable QUBO.

Lemma 3.1 (Non-degradation on a fixed population)**.** *For every $K \ge 1$ and every input tuple $\mathcal{P} = (\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K)})$,*

$$f(\Phi_K(\mathcal{P})) \le \min_{k\in[K]} f(\mathsf{Polish}(\boldsymbol{x}^{(k)})).$$

If the tuple contains the best rounded replica, this also bounds the baseline polish on the same population.

The best polished elite is always available to the incumbent. Relinking can therefore improve its objective or leave it unchanged. This is a paired guarantee for a fixed population and objective; it does not compare independent annealing runs. If $K = 1$, or all selected elites polish to the same state, I-PRP returns exactly the baseline polished state.

Proposition 3.2 (Improvement along a two-step path)**.** *Suppose $K \ge 2$ and the polished states $\boldsymbol{x}^{(1)\star}$ and $\boldsymbol{x}^{(k)\star}$, for some $k \in \{2, \dots, K\}$, disagree on exactly two distinct coordinates $\{i, j\}$. Let $i_0^* = \min \arg\min_{m\in\{i,j\}} \Delta_m(\boldsymbol{x}^{(1)\star})$. If*

$$f(\mathsf{Polish}(\sigma_{i_0^*}\boldsymbol{x}^{(1)\star})) < f(\boldsymbol{x}^{(1)\star}),$$

then $f(\Phi_K(\mathcal{P})) < f(\boldsymbol{x}^{(1)\star})$.

This is the shortest path with a non-endpoint state. Both endpoints are already polished, and the target is no better than the source. The intermediate state can nevertheless initiate a descent to a better basin. The condition concerns the coordinate actually selected by the deterministic rule; it does not assert that every improving two-flip move is explored. Proposition B.1 gives the corresponding statement for longer path prefixes.

3.2 Iterating strict improvements: I-PRP

A successful pass adds a new basin representative to the tuple. Let Sort order a tuple by its raw objective values and lexicographic order. For a sorted tuple $\mathcal{P}$, define

$$\Psi_K(\mathcal{P}) = \begin{cases} \mathrm{Sort}\big(\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K-1)}, \Phi_K(\mathcal{P})\big), & f(\Phi_K(\mathcal{P})) < b(\mathcal{P}), \\ \mathcal{P}, & \text{otherwise.} \end{cases} \tag{11}$$

Thus a strict improvement replaces the worst *raw* member, rather than the worst polished member. Starting from the top-K rounded tuple $\mathcal{P}_0$, iterate $\mathcal{P}_{t+1} = \Psi_K(\mathcal{P}_t)$ until it is unchanged, then return $\Phi_K(\mathcal{P}_t)$. Algorithm 1 gives the complete procedure.

Theorem 3.3 (Finite convergence of I-PRP)**.** *For the exact iteration, let $b_t = b(\mathcal{P}_t)$ and $g_t = f(\Phi_K(\mathcal{P}_t))$. Both sequences are non-increasing. Every update that changes the tuple satisfies*

$$b_{t+1} = g_t < b_t, \qquad g_{t+1} \le g_t.$$

There are at most $|f(\{0,1\}^N)| - 1 \le 2^N - 1$ such updates. The iteration terminates at a tuple $\mathcal{P}^*$, and

$$\Psi_K(\mathcal{P}^*) = \mathcal{P}^* \quad \Longleftrightarrow \quad f(\Phi_K(\mathcal{P}^*)) = b(\mathcal{P}^*).$$

Algorithm 1 Iterative Path-Relinking Polish (I-PRP).

Input: rounded population $\widetilde{\mathcal{P}} \in (\{0,1\}^N)^S$, objective f, integer $1 \le K \le S$
Output: a polished state with objective no greater than polishing the best rounded member

sort $\widetilde{\mathcal{P}}$ by (f, lex); let $\mathcal{Q}$ be its first K entries
loop
  $(\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K)}) \leftarrow \mathcal{Q}$
  compute $\mathsf{Polish}(\boldsymbol{x}^{(k)})$ for all $k \in [K]$
  sort these states by (f, lex), naming them $\boldsymbol{x}^{(1)\star}, \dots, \boldsymbol{x}^{(K)\star}$
  $b \leftarrow f(\boldsymbol{x}^{(1)\star})$; $(\boldsymbol{x}^\star, f^\star) \leftarrow (\boldsymbol{x}^{(1)\star}, b)$
  for $k = 2, \dots, K$ **do**
    $\boldsymbol{c}^{(0,k)} \leftarrow \boldsymbol{x}^{(1)\star}$; $t \leftarrow 0$
    $D \leftarrow \{i \in [N] : c_i^{(0,k)} \neq x_i^{(k)\star}\}$
    while $D \neq \emptyset$ **do**
      $i^* \leftarrow \min\{i \in \arg\min_{j \in D} \Delta_j(\boldsymbol{c}^{(t,k)})\}$
      $\boldsymbol{c}^{(t+1,k)} \leftarrow \sigma_{i^*} \boldsymbol{c}^{(t,k)}$; $D \leftarrow D \setminus \{i^*\}$
      $\boldsymbol{z} \leftarrow \mathsf{Polish}(\boldsymbol{c}^{(t+1,k)})$ ▷ leave the path state unchanged
      if $f(\boldsymbol{z}) < f^\star$ **then**
        $(\boldsymbol{x}^\star, f^\star) \leftarrow (\boldsymbol{z}, f(\boldsymbol{z}))$
      end if
      $t \leftarrow t + 1$
    end while
  end for
  if $f^\star = b$ **then**
    return $(\boldsymbol{x}^\star, f^\star)$
  end if
  $\mathcal{Q} \leftarrow \mathrm{Sort}(\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K-1)}, \boldsymbol{x}^\star)$
end loop

The inserted solution is already polished and is strictly better than every polished member of the old tuple. It therefore becomes the next pass's best polished elite. Since a finite objective range admits only finitely many strict decreases, the loop must stop. The bound proves termination; it is not a polynomial-time complexity guarantee. Lemma 3.1 and the theorem also imply non-degradation relative to the initial baseline throughout the iteration.

Corollary 3.4 (Optimality over the final relinking paths)**.** *At a fixed point $\mathcal{P}^*$, let $\boldsymbol{x}^{(1)\star,*}, \dots, \boldsymbol{x}^{(K)\star,*}$ be its sorted polished elites. For each $k \in \{2, \dots, K\}$, let $d_k^* = H(\boldsymbol{x}^{(1)\star,*}, \boldsymbol{x}^{(k)\star,*})$ and let $\boldsymbol{c}^{(t,k),*}$ be the corresponding deterministic path state. Then*

$$f(\mathsf{Polish}(\boldsymbol{c}^{(t,k),*})) \ge b(\mathcal{P}^*), \qquad 1 \le t \le d_k^*. \tag{12}$$

The final pass has exhausted the routes specified by its elite population: none of their polished states improves the incumbent. This population-covered certificate does not imply optimality over all two-flip or higher-order neighborhoods. Complete proofs appear in Appendix B.

Cost and parallelism. In a dense QUBO, initializing a cached product costs $O(N^2)$; scanning the flip differences, copying a state and cache, or updating them after one flip costs $O(N)$. Let F_k^{elite} and $F_{t,k}^{\mathrm{path}}$ count accepted flips in the respective initial and path-state polish calls. Including worst-case lexicographic sorting, one pass costs

$$O\left(KN^2 + KN\log K + N\sum_{k=1}^{K} F_k^{\mathrm{elite}} + N\sum_{k=2}^{K}\sum_{t=1}^{d_k}(1 + F_{t,k}^{\mathrm{path}})\right). \tag{13}$$

Here $d_k \le N$, and path-polish calls receive copies of the current cache. Sparse matrix updates can use incident nonzeros, although a full coordinate scan still costs $O(N)$ without an additional data structure. Initial elite polishes and different target paths are independent within a pass. They may be batched, with their outputs reduced in the specified order to preserve deterministic ties; steps along an individual path remain sequential.

4 Experiments

We compare the improved sampling algorithm for combinatorial optimization (iSCO) [Sun et al., 2023b, Goshvadi et al., 2023], baseline PQQA [Ichikawa and Arai, 2025], and PQQA followed by I-PRP. The suites include maximum independent set (MIS), maximum clique, and maximum cut (MaxCut) on Erdős–Rényi (ER), Barabási–Albert (BA), and random regular graphs (RRG), as well as established graph benchmarks. In the tables, n is the vertex count and d is the degree of a regular graph. For maximization tasks, the tables report a quality ratio ρ (also called ApR) relative to the stated reference; higher is better. References need not be certified optima, so these ratios can exceed one. Graph coloring instead reports the number of monochromatic edges, where lower is better.

Setup. The benchmark tables use the PQQA annealing protocol, with per-suite settings in Appendix C.2. The core refinement uses $K = 32$ on G-set and $K = 8$ elsewhere, complete paths, and strict improvement. These tables summarize independently run pipelines. Section 4.3 complements them with same-population and matched-runtime controls, directly testing the setting of Lemma 3.1. Synchronized component times are reported in Appendix D.2. Optional configurations and their same-archive ablation appear in Appendices C.4 and D.3, separately from the core operator and its certificate.

Table 1: MIS and maximum-clique benchmark quality from independent runs. Columns report the mean reference ratio $\bar{\rho}$ and standard deviation σ; higher ratios are better. Bold marks the largest displayed mean in each row, including ties; shading identifies I-PRP. RRG ratios use the benchmark reference values [Ichikawa, 2026]. † Reported single-basin baseline value. Paired comparisons and synchronized times appear in Appendix D.

	iSCO		PQQA		PQQA + I-PRP	
Suite	$\bar{\rho}$	σ	$\bar{\rho}$	σ	$\bar{\rho}$	σ
Maximum independent set						
SATLIB-UF ($n\approx1209$)	0.987	0.003	**0.994**	0.002	**0.994**†	0.002
ER $[700, 800]$	0.926	0.014	**0.998**	0.007	0.997	0.008
ER $[9000, 11000]$	0.865	0.007	**0.995**	0.003	**0.995**	0.003
Random regular MIS (dense-Q)						
RRG d=20, n=10^4	0.679	0.001	**0.757**	0.001	0.755	0.001
RRG d=100, n=10^4	0.835	0.004	0.943	0.002	**0.949**	0.003
RRG d=20, n=10^5	0.591	0.001	**0.753**	0.000	**0.753**	0.000
RRG d=100, n=10^5	0.739	0.002	**0.954**	0.001	0.953	0.001
Maximum clique						
Twitter	**1.000**	0.000	0.999	0.009	**1.000**	0.003
RBtest	0.880	0.050	0.879	0.053	**0.888**	0.053

4.1 QUBO suites: MIS, Max Clique, and MaxCut

Maximum independent set. Table 1 reports ratios on SATLIB-UF [Hoos and Stützle, 2000], ER, and regular graphs. The SATLIB-UF result matches baseline PQQA at 0.994. Both PQQA configurations achieve ratios near the reference value on the two ER suites. On the high-degree regular graph with $d = 100$ and $n = 10^4$, I-PRP increases the reported ratio from 0.943 to 0.949. The regular-graph results use the dense QUBO representation and their stated benchmark references.

Maximum clique. On Twitter, I-PRP reaches a displayed mean ratio of 1.000. On RBtest, the ratio increases from 0.879 for baseline PQQA to 0.888, compared with 0.880 for iSCO. RBtest thus provides a setting in which retaining and relinking several polished candidates improves the aggregate solution quality.

Maximum cut. Table 2 reports the core operator on G-set. Its mean cut exceeds baseline PQQA on 15 of the 16 instances and exceeds iSCO on 11, with gains on G55–G57, G59–G62, and G64–G67. For example, the mean cut on G59 increases from 5865.6 to 5966.8, and on G64 from 8372.0 to

8556.0. These improvements show the value of refining a population after the annealing schedule has ended.

Table 2: Mean cut values on G-set [Ye, 2003]; higher is better. These are independent benchmark runs, rather than refinements of a shared population. Raw cuts permit direct comparison without a normalization reference. Bold marks the largest displayed mean in each row.

Instance	iSCO	PQQA	PQQA + I-PRP
G14	**3054.6**	2970.4	3044.6
G15	**3042.8**	2945.0	3035.6
G22	**13312.2**	13119.4	13279.4
G55	10165.0	10195.2	**10209.8**
G56	3873.6	3916.2	**3951.8**
G57	3398.4	3396.0	**3428.0**
G58	**19104.2**	18800.8	19064.8
G59	5875.6	5865.6	**5966.8**
G60	13952.4	14041.2	**14049.0**
G61	5555.4	5660.0	**5684.6**
G62	4719.2	4725.6	**4771.2**
G63	**26749.4**	26343.6	26706.2
G64	8388.6	8372.0	**8556.0**
G65	5356.0	5391.3	**5393.3**
G66	6106.7	6140.7	**6151.3**
G67	6660.0	**6724.7**	6721.3

4.2 Graph coloring with categorical I-PRP

For a simple undirected graph $G = ([N], E)$ and an integer $L \geq 2$, an L-color configuration belongs to $[L]^N$ and the objective is

$$f(\boldsymbol{x}) = \sum_{\{u,v\} \in E} \mathbf{1}\{x_u = x_v\}, \tag{14}$$

where $\mathbf{1}\{A\}$ is 1 when A holds and 0 otherwise. A local move changes a vertex i to a color $a \neq x_i$. The categorical polish repeatedly selects the move with the smallest objective difference, breaks ties lexicographically by (i, a), and accepts a move only when its difference is negative, stopping when none remains. A relinking step instead changes one disagreeing vertex directly to its target color, choosing the smallest objective difference and then the smallest vertex index. Each step removes one disagreement. The finite-descent and idempotence arguments, and hence Lemma 3.1 and Theorem 3.3, apply on the finite state space $[L]^N$, with L^N replacing 2^N in the state-count bound.

Table 3 compares with Tabucol, a tabu-search method, and PI-SAGE, the GraphSAGE variant of the physics-inspired graph neural network solver. I-PRP retains the zero-conflict solutions of PQQA and reaches proper colorings on anna and jean, reducing their reported conflicts from 3 and 1 to zero. The other reported conflict counts are unchanged. This extension uses the same elite selection and strict promotion rule as the binary operator.

4.3 Paired attribution and compute controls

The paired study fixes the rounded PQQA population for each instance and seed, saves its tensor, and verifies its SHA-256 hash across refinement variants. Table 4 compares core I-PRP with standard one-replica polish on 90 population–seed records. It yields 29 strict improvements and 61 ties, directly matching the pointwise setting of Lemma 3.1. The confidence intervals use 10,000 percentile cluster-bootstrap resamples of instances, with all seeds of an instance retained together. Appendix D.1 gives the complete component comparison and a separate paper-scale ER-MaxCut control.

Where the gain comes from. On the 15 G-set pairs, polishing the elite endpoints improves the mean cut by 55.133 over standard polish. Relinking without intermediate re-polishing gives 72.533; one

Table 3: Graph coloring conflicts with L colors; lower is better. Tabucol [Hertz and de Werra, 1987] and PI-SAGE reference scores are from Schuetz et al. [2022a]. Mycielski rows use benchmark-specific identifiers and show only PQQA-based results. L is the tested color budget; dashes denote omitted reference comparisons.

Instance	L	Tabucol	PI-SAGE	PQQA	**I-PRP**
myciel3	3	–	–	0	**0**
myciel4	4	–	–	0	**0**
myciel5	6	–	–	0	**0**
myciel6	7	–	–	0	**0**
myciel7	8	–	–	0	**0**
queen5_5	5	0	0	0	**0**
queen6_6	7	0	0	0	**0**
queen7_7	7	0	0	0	**0**
queen8_8	9	0	1	0	**0**
queen9_9	10	0	1	0	**0**
queen10_10	11	–	–	1	**1**
queen11_11	11	20	17	11	**11**
queen12_12	12	–	–	13	**13**
queen13_13	13	35	26	14	**14**
anna	11	–	–	3	**0**
jean	10	–	–	1	**0**
queen8_12	12	–	–	0	**0**

Table 4: Paired gains of core I-PRP over standard polish on identical rounded populations. Gains are in cut units on G-set and vertices on RBtest and RRG-MIS. "Wins" counts strict gains; all other records are ties. These targeted populations differ from the larger benchmark suites above.

Suite	Pairs	Mean gain	95% confidence interval	Wins
G-set G1–G5	15	$+194.600$	$[176.067, 215.067]$	15/15
RBtest, first 20 instances	60	$+0.167$	$[0.033, 0.317]$	10/60
RRG-MIS ($d = 100$, $n = 1{,}000$)	15	$+0.267$	$[0.067, 0.467]$	4/15

complete re-polishing pass gives 141.667; iteration gives 194.600. These comparisons distinguish the two candidate sources in Eq. (10): a better polished endpoint and a better descent initiated along a path. Reinserting an improvement then changes the paths available to the next pass. On a separate six-population ER study, the core method obtains 84.4% of the gain of the full optional extension stream. Same-archive forward, reverse, and bidirectional relinking controls are reported in Appendix D.3.

Quality under matched time budgets. We use synchronized measurements on an NVIDIA GH200 and set the target budget to shared PQQA annealing followed by core I-PRP. On G1–G5, I-PRP exceeds longer PQQA by 213.533 cuts and restarted PQQA by 115.600 cuts on average, winning all 15 pairs in each comparison. The competitors use 0.970 and 0.966 of the target time. The advantage over longer PQQA is also present at each reported quarter-budget checkpoint. On the targeted RRG-MIS suite, I-PRP exceeds runtime-matched iSCO by 7.000 vertices on average across 15 wins, at an iSCO-to-target time ratio of 0.985. Appendix D.2 reports the timing decomposition, uncertainty, and the RBtest controls.

Population size and returned solutions. The six-population ER sweep gives a mean gain of 20.667 cuts at $K = 8$, compared with 25.000 at $K = 64$, using 1.946 rather than 25.491 seconds of mean post-processing time. Thus a small archive captures most of the gain observed at $K = 64$ in this study. Appendix D.4 separates this fixed-population comparison from reruns that change the upstream replica count or diversity coefficient. Independent checks on 15 ER populations find no improving two-flip or three-flip move in any returned I-PRP state; the corresponding pass counts are 2/15 for standard polish and 5/15 for one pass. These are empirical properties of the tested outputs, while Corollary 3.4 remains a certificate over the final population's paths.

5 Related work

Path relinking and elite archives. Path relinking uses reference solutions to guide trajectories through discrete search spaces [Glover et al., 2000]. Its combinations with GRASP, tabu search, and quadratic optimization are well established [Festa et al., 2002, Resende and Ribeiro, 2005, Ribeiro and Resende, 2012, Martí et al., 2009, Wang et al., 2012b]. Other effective local and population search approaches include breakout search, tabu search, and multilevel refinement [Benlic and Hao, 2013, Kochenberger et al., 2013, Wang et al., 2012a]. Our contribution concerns where the reference set comes from and how the refinement stops: the final rounded PQQA population supplies the archive, and strict improvements drive a deterministic outer iteration. The method does not require an archive to be maintained during annealing.

Training-free optimization. Annealing, Markov chain Monte Carlo, and locally informed samplers explore low-cost states without supervised training [Kirkpatrick et al., 1983, Neal, 1996, Hukushima and Nemoto, 1996, Earl and Deem, 2005, Angelini and Ricci-Tersenghi, 2019, Grathwohl et al., 2021, Zhang et al., 2022]. iSCO is a discrete sampling baseline in the PQQA benchmark protocol [Sun et al., 2023b, Goshvadi et al., 2023]. Continuous relaxations and PQQA instead expose differentiable objectives to parallel computation [Ichikawa, 2024, Ichikawa and Arai, 2025]. I-PRP uses only their final discrete candidates, a scalar objective, and a deterministic local polish; the upstream sampling or relaxation dynamics do not enter its guarantees.

Learning-based solvers. Graph neural networks and neural relaxations learn or optimize heuristics for combinatorial tasks [Li et al., 2018, Dai et al., 2017, Karalias and Loukas, 2020, Schuetz et al., 2022b, Sun et al., 2022, Xu et al., 2020, Cappart et al., 2023, Bengio et al., 2021]. Studies of physics-inspired graph neural networks also emphasize comparisons with established search methods [Angelini and Ricci-Tersenghi, 2023, Boettcher, 2023, Schuetz et al., 2023]. I-PRP requires no model training. Its interface could also refine a population generated by a learned solver, provided the objective and deterministic polish are supplied.

6 Conclusion

The final population of a parallel annealer can support further search after its best candidate has been selected. I-PRP uses polished elites to define relinking paths and new local descents, retaining a new solution only under strict improvement. For the same rounded input, it does not worsen the baseline objective, terminates in a finite number of updates, and exhausts the polished candidates on the final population's paths. The QUBO and categorical results illustrate how this refinement can improve solutions while leaving the annealing loop unchanged.

Scope and limitations. I-PRP depends on the supplied objective, penalty weights, and population. Relinking provides no gain beyond endpoint polishing if the elites share one basin or their paths expose no improving descent. Refinement adds problem-dependent computation; the elite-size sweep and synchronized timings quantify this quality–time trade-off. Its certificate concerns only the final elite-induced paths and guarantees neither strict improvement nor global optimality. The finite-state upper bounds are exponential. Optional candidate generators and truncated-path variants are evaluated separately from the complete-path certificate.

Broader impact. Improved combinatorial search may support scheduling and resource allocation. Its downstream effects depend on the chosen objective and constraints: optimizing an incomplete model can reinforce undesirable allocation decisions. The method itself does not assess fairness or the suitability of a deployment objective.

Acknowledgments and Disclosure of Funding

Funding. The author received no external funding or other third-party support for this work.

Competing interests. The author declares no competing interests.

References

Maria Chiara Angelini and Federico Ricci-Tersenghi. Monte Carlo algorithms are very effective in finding the largest independent set in sparse random graphs. *Physical Review E*, 100(1):013302, 2019.

Maria Chiara Angelini and Federico Ricci-Tersenghi. Modern graph neural networks do worse than classical greedy algorithms in solving combinatorial optimization problems like maximum independent set. *Nature Machine Intelligence*, 5(1):29–31, 2023.

Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

Una Benlic and Jin-Kao Hao. Breakout local search for the max-cut problem. *Engineering Applications of Artificial Intelligence*, 26(3):1162–1173, 2013.

Stefan Boettcher. Inability of a graph neural network heuristic to outperform greedy algorithms in solving combinatorial optimization problems. *Nature Machine Intelligence*, 5(1):24–25, 2023.

Quentin Cappart, Didier Chételat, Elias B. Khalil, Andrea Lodi, Christopher Morris, and Petar Veličković. Combinatorial optimization and reasoning with graph neural networks. *Journal of Machine Learning Research*, 24(130):1–61, 2023. URL https://jmlr.org/papers/v24/21-0449.html.

Hanjun Dai, Elias B. Khalil, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/7214-learning-combinatorial-optimization-algorithms-over-graphs.

David J Earl and Michael W Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

Paola Festa, Panos M Pardalos, Mauricio GC Resende, and Celso C Ribeiro. Randomized heuristics for the max-cut problem. *Optimization Methods and Software*, 17(6):1033–1058, 2002.

Fred Glover, Manuel Laguna, and Rafael Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 29(3):653–684, 2000.

Katayoon Goshvadi, Haoran Sun, Xingchao Liu, Azade Nova, Ruqi Zhang, Will Grathwohl, Dale Schuurmans, and Hanjun Dai. DISCS: A benchmark for discrete sampling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 79035–79066. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f9ad87c1ebbae8a3555adb31dbcacf44-Paper-Datasets_and_Benchmarks.pdf.

Will Grathwohl, Kevin Swersky, Milad Hashemi, David Duvenaud, and Chris Maddison. Oops I took a gradient: Scalable sampling for discrete distributions. In *International Conference on Machine Learning*, pages 3831–3841. PMLR, 2021.

Alain Hertz and Dominique de Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, 1987. doi: 10.1007/BF02239976.

Holger H. Hoos and Thomas Stützle. SATLIB: An online resource for research on SAT. In Ian P. Gent, Hans van Maaren, and Toby Walsh, editors, *SAT 2000*, pages 283–292. IOS Press, 2000. URL https://www.cs.ubc.ca/~hoos/SATLIB/.

Koji Hukushima and Koji Nemoto. Exchange Monte Carlo method and application to spin glass simulations. *Journal of the Physical Society of Japan*, 65(6):1604–1608, 1996.

Yuma Ichikawa. Controlling continuous relaxation for combinatorial optimization. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ykACV1IhjD.

Yuma Ichikawa. QQA4CO combinatorial optimization benchmark suite. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/Yuma-Ichikawa/qqa4co-bench. Dataset documentation accessed November 7, 2026.

Yuma Ichikawa and Yamato Arai. Optimization by parallel quasi-quantum annealing with gradient-based sampling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9EfBeXaXf0.

Nikolaos Karalias and Andreas Loukas. Erdős goes neural: An unsupervised learning framework for combinatorial optimization on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pages 6659–6672, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/49f85a9ed090b20c8bed85a5923c669f-Abstract.html.

Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972. doi: 10.1007/978-1-4684-2001-2_9.

Scott Kirkpatrick, C. Daniel Gelatt, Jr., and Mario P Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

Gary Kochenberger, Jin-Kao Hao, Zhipeng Lü, Haibo Wang, and Fred Glover. Solving large scale max cut problems via tabu search. *Journal of Heuristics*, 19(4):565–571, 2013.

Gary Kochenberger, Jin-Kao Hao, Fred Glover, Mark Lewis, Zhipeng Lü, Haibo Wang, and Yang Wang. The unconstrained binary quadratic programming problem: a survey. *Journal of Combinatorial Optimization*, 28:58–81, 2014.

Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer, fifth edition, 2012. doi: 10.1007/978-3-642-24488-9.

Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Combinatorial optimization with graph convolutional networks and guided tree search. In *Advances in Neural Information Processing Systems*, volume 31, pages 539–548, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/8d3bba7425e7c98c50f52ca1b52d3735-Abstract.html.

Helena R Lourenço, Olivier C Martin, and Thomas Stützle. Iterated local search. In *Handbook of metaheuristics*, pages 320–353. Springer, 2003.

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

Rafael Martí, Abraham Duarte, and Manuel Laguna. Advanced scatter search for the max-cut problem. *INFORMS Journal on Computing*, 21(1):26–38, 2009.

Peter Merz and Bernd Freisleben. Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of Heuristics*, 8(2):197–213, 2002.

Radford M Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6:353–366, 1996.

Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publications, 1998.

Mauricio G. C. Resende and Celso C. Ribeiro. GRASP with path-relinking: Recent advances and applications. In *Metaheuristics: Progress as Real Problem Solvers*, pages 29–63. Springer, 2005.

Celso C Ribeiro and Mauricio GC Resende. Path-relinking intensification methods for stochastic local search algorithms. *Journal of Heuristics*, 18(2):193–214, 2012.

Martin J. A. Schuetz, J. Kyle Brubaker, Zhihuai Zhu, and Helmut G. Katzgraber. Graph coloring with physics-inspired graph neural networks. *Physical Review Research*, 4(4):043131, 2022a. doi: 10.1103/PhysRevResearch.4.043131.

Martin JA Schuetz, J Kyle Brubaker, and Helmut G Katzgraber. Combinatorial optimization with physics-inspired graph neural networks. *Nature Machine Intelligence*, 4(4):367–377, 2022b.

Martin JA Schuetz, J Kyle Brubaker, and Helmut G Katzgraber. Reply to: Modern graph neural networks do worse than classical greedy algorithms in solving combinatorial optimization problems like maximum independent set. *Nature Machine Intelligence*, 5(1):32–34, 2023.

Haoran Sun, Etash K Guha, and Hanjun Dai. Annealed training for combinatorial optimization on graphs. *arXiv preprint arXiv:2207.11542*, 2022.

Haoran Sun, Hanjun Dai, Bo Dai, Haomin Zhou, and Dale Schuurmans. Discrete Langevin samplers via Wasserstein gradient flow. In *International Conference on Artificial Intelligence and Statistics*, pages 6290–6313. PMLR, 2023a.

Haoran Sun, Katayoon Goshvadi, Azade Nova, Dale Schuurmans, and Hanjun Dai. Revisiting sampling for combinatorial optimization. In *International Conference on Machine Learning*, pages 32859–32874. PMLR, 2023b.

Yang Wang, Zhipeng Lü, Fred Glover, and Jin-Kao Hao. A multilevel algorithm for large unconstrained binary quadratic optimization. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 7298 of *Lecture Notes in Computer Science*, pages 395–408. Springer, 2012a. doi: 10.1007/978-3-642-29828-8_26.

Yang Wang, Zhipeng Lü, Fred Glover, and Jin-Kao Hao. Path relinking for unconstrained binary quadratic programming. *European Journal of Operational Research*, 223(3):595–604, 2012b. doi: 10.1016/j.ejor.2012.07.012.

Hao Xu, Ka-Hei Hui, Chi-Wing Fu, and Hao Zhang. TilinGNN: Learning to tile with self-supervised graph neural network. *ACM Transactions on Graphics*, 39(4):129:1–129:16, 2020. doi: 10.1145/3386569.3392380. SIGGRAPH 2020.

Yinyu Ye. The G-set max-cut benchmark instances. `https://web.stanford.edu/~yyye/yyye/Gset/`, 2003. Online benchmark suite.

Ruqi Zhang, Xingchao Liu, and Qiang Liu. A Langevin-like sampler for discrete distributions. In *International Conference on Machine Learning*, pages 26375–26396. PMLR, 2022.

A Algorithmic Details

Algorithm 1 gives the exact iteration in the main paper. The following example tracks all its decisions, including the path-state copy, the raw-rank replacement, and the final stopping test.

A.1 A complete three-variable example

Consider the QUBO

$$f(\boldsymbol{x}) = 2x_1 + 3x_2 + 2x_3 - 4x_1x_2 - 5x_1x_3 + 4x_2x_3, \qquad \boldsymbol{x} \in \{0,1\}^3,$$

and the initial elite tuple $\mathcal{P}_0 = (000, 110)$, writing binary triples without commas. Both elites are strict one-flip local minima: their objective values are 0 and 1, and their flip differences are $(2, 3, 2)$ and $(2, 1, 1)$, respectively. Stage 1 therefore leaves them unchanged. The target 110 differs from 000 in coordinates 1 and 2. Since their initial flip costs are 2 and 3, the first relinking path is

$$000\ (f = 0)\ \longrightarrow\ 100\ (f = 2)\ \longrightarrow\ 110\ (f = 1).$$

At 100, the flip differences are $(-2, -1, -3)$, so polishing flips coordinate 3 and reaches 101 with objective -1. Its flip differences are $(3, 3, 3)$, so it is also a strict local minimum. Thus $\Phi_2(\mathcal{P}_0) = 101$: polishing leaves the relinking path by changing a coordinate on which the endpoints agree.

The update replaces the worst raw elite 110, giving $\mathcal{P}_1 = (101, 000)$. The second pass relinks these two states. Both differing coordinates have flip cost 3, so the index rule selects coordinate 1:

$$101\ (f = -1)\ \longrightarrow\ 001\ (f = 2)\ \longrightarrow\ 000\ (f = 0).$$

Polishing 001 returns 101, because its flip differences are $(-3, 7, -2)$; polishing 000 leaves it unchanged. Neither candidate improves the incumbent value -1. Hence $\mathcal{P}_1$ is a fixed point and I-PRP returns 101 after one strict population update.

B Proofs

Throughout, $f : \{0,1\}^N \to \mathbb{R}$ is fixed, comparisons are exact, and every relinking path is complete. The smallest-index rule makes Polish deterministic. Each accepted flip strictly decreases f, so the finite state space ensures termination. The stopping rule implies one-flip local optimality and, consequently, idempotence. For a tuple $\mathcal{P}$, write $b(\mathcal{P}) = f(\boldsymbol{x}^{(1)\star})$ and $d_k = H(\boldsymbol{x}^{(1)\star}, \boldsymbol{x}^{(k)\star})$. The state $\boldsymbol{c}^{(t,k)}$ is the t-th state on the path to elite k; all endpoints remain fixed during a pass.

B.1 Proof of Lemma 3.1

Proof. The incumbent starts at the smallest polished-elite value $b(\mathcal{P})$ and is replaced precisely when a candidate has a strictly smaller value. Induction over the finite sequence of evaluated candidates proves Eq. (10). The first set is nonempty even when there are no path states. Since sorting does not change the polished values,

$$f(\Phi_K(\mathcal{P})) \le b(\mathcal{P}) = \min_{1 \le k \le K} f(\mathsf{Polish}(\boldsymbol{x}^{(k)})).$$

For the top-K prefix of the same sorted rounded population, $\boldsymbol{x}^{(1)}$ is the upstream choice, yielding the paired non-degradation claim. Every possible returned state is a polish output and hence is itself polished. □

B.2 Proof of Proposition 3.2

Proof. Fix the elite k in the hypothesis. Its disagreement set is $\{i, j\}$, so the deterministic path rule gives $\boldsymbol{c}^{(1,k)} = \sigma_{i_0^*}\boldsymbol{x}^{(1)\star}$. Stage 3 evaluates this state, and Eq. (10) implies

$$f(\Phi_K(\mathcal{P})) \le f\big(\mathsf{Polish}(\sigma_{i_0^*}\boldsymbol{x}^{(1)\star})\big) < f(\boldsymbol{x}^{(1)\star}).$$

The endpoint is already polished and has value at least $f(\boldsymbol{x}^{(1)\star})$ by sorting. Thus the asserted improvement is supplied by the polished intermediate, or by an even better candidate evaluated elsewhere in the pass. □

B.3 Proof of Theorem 3.3

Proof. Write $\mathcal{P}_t = (\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(K)})$, and let $b_t = b(\mathcal{P}_t)$ and $g_t = f(\Phi_K(\mathcal{P}_t))$. Lemma 3.1 gives $g_t \le b_t$. The returned state $\boldsymbol{y}_t = \Phi_K(\mathcal{P}_t)$ is polished, so

$$\mathsf{Polish}(\boldsymbol{y}_t) = \boldsymbol{y}_t, \qquad f(\mathsf{Polish}(\boldsymbol{y}_t)) = g_t. \tag{15}$$

If $g_t = b_t$, the tuple is unchanged, and determinism gives $b_{t+1} = b_t$ and $g_{t+1} = g_t$.

Suppose $g_t < b_t$. Every retained raw member has polished value at least b_t, whereas the inserted member has polished value g_t. Therefore

$$b_{t+1} = \min\Big(\{g_t\} \cup \{f(\mathsf{Polish}(\boldsymbol{x}_t^{(k)})) : 1 \le k < K\}\Big) = g_t < b_t. \tag{16}$$

Moreover, $\boldsymbol{y}_t$ cannot already be a member of $\mathcal{P}_t$: idempotence would then give a polished-elite value $g_t < b_t$, contradicting the definition of b_t. Thus insertion changes the tuple even after sorting. Applying Lemma 3.1 to the next tuple yields

$$g_{t+1} \le b_{t+1} = g_t.$$

Both sequences are consequently non-increasing, with strict decrease of b_t at every update.

The set $f(\{0,1\}^N)$ has at most 2^N elements. Hence there are at most $|f(\{0,1\}^N)| - 1$ updates. Each pass also terminates, since it contains finitely many path moves and terminating polish calls. The iteration therefore reaches a fixed tuple $\mathcal{P}^*$. For any sorted tuple, $f(\Phi_K(\mathcal{P})) \le b(\mathcal{P})$; equality selects the unchanged branch, while strict inequality changes the tuple as just proved. This establishes the claimed fixed-point characterization in both directions. □

B.4 Proof of Corollary 3.4

Proof. At a fixed tuple, Theorem 3.3 gives $f(\Phi_K(\mathcal{P}^*)) = b(\mathcal{P}^*)$. By Eq. (10), every evaluated candidate must then have value at least $b(\mathcal{P}^*)$, which is exactly Eq. (12). □

B.5 Longer relinking paths

Proposition B.1 (Improvement from a path prefix)**.** *Let $K \ge 2$ and $k \in \{2, \ldots, K\}$. Suppose the polished elites $\boldsymbol{x}^{(1)\star}$ and $\boldsymbol{x}^{(k)\star}$ differ on a set $D \subseteq [N]$ of size $m \ge 2$. Let $\pi : [m] \to D$ be the bijection listing the coordinates selected, in order, by their deterministic Stage 2 path. For $1 \le r < m$, define*

$$\boldsymbol{y}^{(r)} = \sigma_{\pi(r)} \cdots \sigma_{\pi(1)} \boldsymbol{x}^{(1)\star}. \tag{17}$$

If, for some such r,

$$f\big(\mathsf{Polish}(\boldsymbol{y}^{(r)})\big) < f(\boldsymbol{x}^{(1)\star}), \tag{18}$$

then $f(\Phi_K(\mathcal{P})) < f(\boldsymbol{x}^{(1)\star})$.

Proof. Each path move flips one coordinate that disagrees with the target, making that coordinate agree without changing any other coordinate. Thus the path selects every element of D exactly once. Induction on t gives

$$\boldsymbol{c}^{(t,k)} = \sigma_{\pi(t)} \cdots \sigma_{\pi(1)} \boldsymbol{x}^{(1)\star}, \qquad D_t^{(k)} = D \setminus \{\pi(1), \ldots, \pi(t)\}, \tag{19}$$

for $1 \le t \le m$. In particular, $\boldsymbol{c}^{(r,k)} = \boldsymbol{y}^{(r)}$. Equation (10) and the hypothesis imply

$$f(\Phi_K(\mathcal{P})) \le f\big(\mathsf{Polish}(\boldsymbol{y}^{(r)})\big) < f(\boldsymbol{x}^{(1)\star}),$$

as required. □

At $t = m$, the path reaches $\boldsymbol{x}^{(k)\star}$, whose polished value is at least that of $\boldsymbol{x}^{(1)\star}$. Only a proper prefix can therefore satisfy Eq. (18). The prefix itself may be uphill; the improvement is obtained by polishing it. Proposition 3.2 is the case $m = 2$ and $r = 1$.

C Additional Implementation Details

This appendix gives the local objective updates, experimental settings, and optional candidate generators.

C.1 One-flip differences and cache updates

For a symmetric QUBO objective $f(\boldsymbol{x}) = \boldsymbol{x}^\top Q\boldsymbol{x} + \boldsymbol{q}^\top \boldsymbol{x}$, let $\boldsymbol{e}_i$ be the i-th standard basis vector, set $d_i = 1 - 2x_i \in \{-1, 1\}$, and write $\boldsymbol{x}' = \sigma_i \boldsymbol{x} = \boldsymbol{x} + d_i \boldsymbol{e}_i$. Expanding the objective difference gives

$$\begin{aligned}\Delta_i^{\mathrm{QUBO}}(\boldsymbol{x}) &= (\boldsymbol{x} + d_i\boldsymbol{e}_i)^\top Q(\boldsymbol{x} + d_i\boldsymbol{e}_i) + \boldsymbol{q}^\top(\boldsymbol{x} + d_i\boldsymbol{e}_i) - \boldsymbol{x}^\top Q\boldsymbol{x} - \boldsymbol{q}^\top\boldsymbol{x} \\ &= d_i\boldsymbol{e}_i^\top Q\boldsymbol{x} + d_i\boldsymbol{x}^\top Q\boldsymbol{e}_i + d_i^2 Q_{ii} + d_i q_i \\ &= 2d_i(Q\boldsymbol{x})_i + Q_{ii} + d_i q_i, \end{aligned} \tag{20}$$

where symmetry of Q and $d_i^2 = 1$ are used in the last line. The formula in the main text follows by writing

$$\begin{aligned}&(1 - 2x_i)\big[q_i + Q_{ii} + 2(Q\boldsymbol{x})_i - 2Q_{ii}x_i\big] \\ &\qquad = d_i q_i + 2d_i(Q\boldsymbol{x})_i + d_i Q_{ii}(1 - 2x_i) = d_i q_i + 2d_i(Q\boldsymbol{x})_i + Q_{ii}. \end{aligned} \tag{21}$$

When the linear term is folded into the diagonal by $Q_{ii} \leftarrow Q_{ii} + q_i$ and $\boldsymbol{q} = 0$, this reduces to the diagonal-folded formula used in many QUBO implementations. A non-symmetric matrix can be replaced by $(Q + Q^\top)/2$ without changing $\boldsymbol{x}^\top Q\boldsymbol{x}$, so the symmetry assumption is not restrictive for QUBO objectives.

For Ising objectives $f(\boldsymbol{s}) = -1/2\boldsymbol{s}^\top J\boldsymbol{s} - \boldsymbol{h}^\top\boldsymbol{s}$ with symmetric J and $J_{ii} = 0$, flipping spin s_i changes only terms incident to i. Since the interaction contribution involving i is $-s_i \sum_{j\neq i} J_{ij}s_j$, the flip changes it to $+s_i \sum_{j\neq i} J_{ij}s_j$. The field term changes from $-h_i s_i$ to $+h_i s_i$. Thus

$$\Delta_i^{\mathrm{spin}}(\boldsymbol{s}) = 2s_i \sum_{j\neq i} J_{ij}s_j + 2h_i s_i = 2s_i\big[(J\boldsymbol{s})_i + h_i\big]. \tag{22}$$

The cached products are updated in $O(N)$ per accepted flip in dense form. If $\boldsymbol{x}' = \boldsymbol{x} + d_i\boldsymbol{e}_i$, then $Q\boldsymbol{x}' = Q\boldsymbol{x} + d_i Q\boldsymbol{e}_i$. If spin i is flipped, then $J\boldsymbol{s}' = J\boldsymbol{s} - 2s_i J\boldsymbol{e}_i$, using the spin value before the flip. Thus a polish call needs one initial cached product, then scans all one-flip differences and updates the product after each accepted flip [Merz and Freisleben, 2002].

C.2 Experimental configuration and reproduction essentials

The original benchmark runs follow the PQQA hyperparameters of Ichikawa and Arai [2025]. Their default settings use AdamW with learning rate 0.1, entropy exponent $\alpha = 4$, Langevin temperature $T = 10^{-3}$, and communication-term coefficient $\alpha_{\mathrm{div}} = 0.2$. For these benchmarks, I-PRP uses $K = 32$ on G-set and $K = 8$ elsewhere, with complete paths and strict improvement. The paired protocols and sensitivity variations are specified separately in Appendix D.

Table 5: PQQA hyperparameters per problem class.

Suite	S (replicas)	Steps	$\gamma_{\min}$	$\gamma_{\max}$
MaxCut Optsicom, ER, BA, G14, G15	1000	30000	-5	0.1
MaxCut G-set G22 (n=2,000)	400	30000	-20	0.1
MaxCut G-set G55 (n=5,000)	200	30000	-20	0.1
MIS SATLIB-UF, MIS ER ($n \in [700, 800]$)	1000	30000	-2	0.1
MIS RRG ($n = 10^4, d \in \{20, 100\}$)	200	30000	-2	0.1
Max Clique Twitter, RBtest	1000	30000	-2	0.1

The benchmark tables report aggregate quality scores; Table 1 also gives the reported standard deviations. QUBO and coloring suites use the seed sets and upstream benchmark splits of Ichikawa and Arai [2025]. For batched suites with a common maximum dimension, instances are grouped along the instance axis; G-set and the $n = 10^4$ regular-graph MIS suite use a single-instance pipeline because their dimensions vary substantially. The additional paired studies in Appendix D use saved populations and explicitly reported sample counts. Their synchronized component timings separate post-processing cost from the shared anneal.

C.3 Optional dense-MaxCut extensions

The formal results in Section 3 apply to the bare deterministic Φ_K/Ψ_K operator. The MaxCut configurations in Table 6 enable an optional guarded extension stack. Each generated candidate is promoted only if it strictly improves the incumbent, but the extended candidate stream is not part of Theorem 3.3 or Corollary 3.4.

Multi-flip path relinking. The optional path step changes one or two disagreeing coordinates at once. For distinct coordinates i, j in a symmetric QUBO, with $d_i = 1 - 2x_i$ and $d_j = 1 - 2x_j$, its two-flip difference is

$$f(\sigma_i\sigma_j\boldsymbol{x}) - f(\boldsymbol{x}) = \Delta_i(\boldsymbol{x}) + \Delta_j(\boldsymbol{x}) + 2d_id_jQ_{ij}.$$

This follows by expanding the quadratic form at $\boldsymbol{x} + d_i\boldsymbol{e}_i + d_j\boldsymbol{e}_j$. Candidate promotion still requires strict improvement of the incumbent.

Consensus-guided subproblem solving. Coordinates on which all elites agree are fixed at their shared values. Substituting those values into the QUBO yields a quadratic objective on the remaining coordinates. For a sufficiently small remaining set, all binary assignments are enumerated and the best candidate is promoted only if it improves the incumbent.

Guarded restart and tabu candidates. Additional candidates are obtained from perturbed population members and tabu-search moves involving up to two variables. They are used in the optional MaxCut configurations and the extension stream in Appendix D.3, with the same strict incumbent guard.

C.4 Benchmark results with optional extensions

Table 6 records the MaxCut configurations that use the optional candidate generators of Appendix C.3. They are separate from the core comparisons in the main text. These entries come from independent end-to-end runs; their elapsed times do not measure the incremental overhead of adding I-PRP to a shared anneal. Table 8 supplies that component comparison, and Table 11 isolates the extension stream on a common archive.

Table 6: MaxCut benchmark configurations with optional extensions. Columns report mean reference ratio $\bar{\rho}$, standard deviation σ, and recorded mean end-to-end time t. Bold marks the largest displayed ratio. The extended PQQA configuration includes candidates beyond the theorem-covered operator. DISCS reference scores are not certified optima [Goshvadi et al., 2023].

	iSCO			PQQA			PQQA + extensions		
Suite	$\bar{\rho}$	σ	t	$\bar{\rho}$	σ	t	$\bar{\rho}$	σ	t
Optsicom	**1.000**	0.000	1.1m	0.999	0.004	13.2s	**1.000**	0.003	15.0s
ER n=256–300	**1.018**	0.017	1.1m	0.926	0.020	1.2m	1.017	0.017	29.9s
BA n=256–300	**1.000**	0.000	1.3m	0.988	0.004	1.2m	0.999	0.002	50.5s
ER n=512–600	**1.002**	0.001	3.9s	0.883	0.003	20.5s	1.000	0.002	11.5s
BA n=512–600	**1.010**	0.005	4.0s	0.982	0.006	20.5s	1.007	0.006	46.4s
ER n=1024–1100	**1.002**	0.001	6.4s	0.903	0.002	22.5s	1.000	0.001	23.8s
BA n=1024–1100	**1.017**	0.003	6.6s	0.982	0.004	22.4s	1.014	0.004	1.6m

D Additional Experiments

D.1 Shared populations and uncertainty

The paired experiments hold the rounded PQQA population fixed for each instance and seed. The effective population tensor is saved and checked by its SHA-256 hash before each post-processing comparison. We distinguish the standard one-replica polish (M_0), the best of the polished top-K endpoints (M_1), cost-guided relinking without intermediate re-polishing (M_2), one complete relinking-and-polish pass (M_3), the iterative core I-PRP operator (M_4), and the optional guarded extension stream (M_5). Thus M_3 evaluates Φ_K once, while M_4 repeats the strict-improvement updates until termination. The fixed-point certificate concerns M_4.

Gains are paired differences in the native maximization score: cut value for MaxCut, clique size for RBtest, and independent-set size for RRG-MIS. A positive difference favors I-PRP. The reported 95% confidence intervals use the percentile cluster bootstrap with $10{,}000$ resamples of instances, retaining all seeds within each sampled instance. Win/tie/loss counts compare paired final scores.

Table 7: Post-processing gains on the same rounded populations, relative to M_0. G-set uses G1–G5; RBtest uses its first 20 instances; RRG-MIS uses $d = 100$ and $n = 1{,}000$. The final interval is for the mean M_4 gain, and "wins" counts its strict gains over M_0.

Suite	Pairs	M_1	M_2	M_3	M_4	Wins	95% CI
G-set	15	$+55.133$	$+72.533$	$+141.667$	$+194.600$	15/15	$[176.067, 215.067]$
RBtest	60	$+0.000$	$+0.083$	$+0.167$	$+0.167$	10/60	$[0.033, 0.317]$
RRG-MIS	15	$+0.000$	$+0.000$	$+0.267$	$+0.267$	4/15	$[0.067, 0.467]$

Table 7 contains 90 matched population–seed records. I-PRP improves 29 and preserves the baseline score in the other 61. On G-set, the progression from endpoint selection to relinking, intermediate re-polishing, and iteration isolates the contribution of each component. A separate six-pair ER-MaxCut comparison uses $S = 1{,}000$, $30{,}000$ PQQA steps, and $K = 8$: M_4 improves on M_0 by 16.667 cuts on average, with six wins and a 95% interval of $[4.667, 29.833]$.

D.2 Synchronized timing and matched-runtime controls

Component timings use an NVIDIA GH200 and the same PQQA anneal for all post-processing variants. GPU components are bracketed by synchronization calls, and warm-up is excluded. Table 8 separates the shared anneal and rounding from the standard polish and the additional core refinement. The entries are component medians, rather than differences between independently timed pipelines.

Table 8: Synchronized component times in seconds. Each entry is a median on the paired suites of Table 7.

Suite	Anneal and round	Standard polish	Additional I-PRP
G-set	1.480	0.04280	62.600
RBtest	1.446	0.00118	0.145
RRG-MIS	1.493	0.00178	2.051

For the matched-runtime comparisons, the target is the time for the shared PQQA anneal followed by core I-PRP. Longer PQQA runs, independent PQQA restarts, and iSCO are calibrated separately on the same hardware. Table 9 reports the actual competitor time divided by this target. The ratios make the extent of budget matching explicit; the RBtest restart control uses 0.893 of the target time.

Table 9: I-PRP relative to runtime-matched controls. Gains use the native units of Table 7. W/T/L counts I-PRP wins, ties, and losses. r_t is competitor time divided by target time; dashes indicate intervals not reported for the iSCO comparisons.

Suite	Control	Mean gain	95% CI	W/T/L	r_t
G-set	Longer PQQA	$+213.533$	$[187.667, 237.533]$	15/0/0	0.970
G-set	PQQA restarts	$+115.600$	$[97.600, 130.067]$	15/0/0	0.966
RBtest	Longer PQQA	$+0.083$	$[-0.033, 0.217]$	11/43/6	0.973
RBtest	PQQA restarts	$+0.167$	$[0.033, 0.317]$	10/50/0	0.893
RRG-MIS	iSCO	$+7.000$	–	15/0/0	0.985
RBtest	iSCO	$+0.067$	–	9/46/5	0.973

On G-set, I-PRP exceeds longer PQQA at all four measured budget checkpoints (Table 10).

Table 10: Best-so-far cut differences between I-PRP and longer PQQA on G1–G5 at fractions of the target budget. The 15 pairs are the same as in Table 9.

Budget	Mean gain	95% CI	W/T/L
25%	$+155.867$	$[135.667, 176.067]$	15/0/0
50%	$+190.400$	$[169.267, 209.337]$	15/0/0
75%	$+204.467$	$[181.400, 223.333]$	15/0/0
100%	$+213.533$	$[187.667, 237.533]$	15/0/0

D.3 Relinking controls and optional extensions

A six-population ER-MaxCut study uses edge probability $p = 0.1$, $n \in \{100, 200\}$, and three seeds per size. Every method in Table 11 receives the same saved archive. The directional controls perform cost-guided relinking with final polishing only; the bidirectional control considers both directions. Core I-PRP additionally re-polishes intermediate states and iterates after a strict gain.

Table 11: Same-archive ER-MaxCut controls. Gains are relative to standard polish M_0; times are mean post-processing times. The final row includes optional candidate generators outside the core fixed-point certificate.

Post-processing method	Mean cut gain	Time (s)
Endpoint polish only (M_1)	$+6.333$	0.028
Forward relinking, final polish	$+15.167$	0.221
Reverse relinking, final polish	$+15.000$	0.215
Bidirectional relinking, final polish	$+16.500$	0.401
Core iterative I-PRP (M_4)	$+20.667$	1.937
Guarded extension stream (M_5)	$+24.500$	2.754

Core I-PRP attains 84.4% of the full extension stream's gain using 70.3% of its post-processing time in this experiment. The directional comparisons also show an additional gain from the complete I-PRP design when the input archive is held fixed.

D.4 Elite size and upstream sensitivity

Table 12 varies K on the same six saved ER populations. Increasing K exposes more potential endpoints, with a corresponding increase in refinement time. Here $K = 8$ yields 82.7% of the gain at $K = 64$ while taking 7.6% of its post-processing time.

Table 12: Elite-size sensitivity on six fixed ER-MaxCut populations. Gains are relative to M_0; each value of K is a separate core I-PRP run on the same saved inputs.

K	Mean cut gain	Strict gains	Mean time (s)
1	0.000	0/6	0.008
2	+15.833	6/6	0.190
4	+18.667	6/6	0.745
8	+20.667	6/6	1.946
16	+24.333	6/6	6.099
32	+24.333	6/6	15.425
64	+25.000	6/6	25.491

Table 13 separates path caps from upstream changes. The cap $T_{\max}$ limits the relinking steps per path; the certificate in Corollary 3.4 covers complete paths. The upstream sweeps rerun PQQA at each population size S or diversity coefficient. Every tested setting improves its paired baseline.

Table 13: Post-processing and true-upstream sensitivity. Fixed-archive settings use six paired records; upstream settings use three. Mean gains are in cut units relative to the paired standard polish.

Parameter	Value	Mean gain	Strict gains
Fixed saved populations			
Path cap $T_{\max}/N$	0.25	+20.000	6/6
	0.50	+20.667	6/6
	1.00	+20.667	6/6
PQQA rerun at each setting			
Population size S	8	+14.000	3/3
	16	+14.000	3/3
	32	+7.000	3/3
	64	+9.667	3/3
Diversity coefficient α_{div}	0.00	+9.667	3/3
	0.05	+7.667	3/3
	0.10	+7.333	3/3
	0.20	+9.667	3/3
	0.40	+12.333	3/3

D.5 Fixed-point diagnostics

In a small exact-enumeration study with 45 populations at $N \in \{12, 14, 16\}$, core I-PRP reaches a global optimum in all 45 cases and improves the standard polish in 22. A separate set of 15 ER-MaxCut populations at $N \in \{100, 200, 300\}$ is checked exhaustively for improving two-flip and three-flip moves. Table 14 reports the number of returned states passing each check. These are empirical neighborhood tests, separate from the population-covered guarantee.

Table 14: Exhaustive neighborhood diagnostics on 15 ER-MaxCut populations. An entry counts returned states with no strictly improving move of the stated size.

Check	Standard polish M_0	One pass M_3	Iterated M_4
Two-flip	2/15	5/15	15/15
Three-flip	2/15	5/15	15/15

The diagnostics also record distinct rounded states and polished basins, Hamming distances, path lengths, improving intermediates, and strict outer updates. The sufficient conditions in Propositions 3.2 and B.1 describe when polishing an intermediate state yields a better basin.

AI assistance. AI tools assisted the experimental workflow, manuscript preparation, and schematic Figure 1. They are not part of the optimization operator or its guarantees.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the post-processing contribution, paired non-degradation guarantee, and scope of the empirical evaluation. The population-dependent certificate is stated in Section 3.2.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses dependence on the upstream population, problem-dependent refinement cost, and the scope of the fixed-point certificate. Appendix D.4 quantifies the elite-size trade-off.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 3 defines the operators and states the assumptions of the main results. Appendix B provides their proofs and the path-prefix result.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: Algorithm 1 specifies the operator, and Appendices C.2 and D describe benchmark settings, paired comparisons, and uncertainty estimation. The paper does not fully specify every ancillary sweep and optional-extension configuration.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper provides pseudocode and cites public benchmark sources. It does not link a versioned end-to-end reproduction package with exact commands for the reported experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: Appendix C.2 gives the optimizer, annealing parameters, and available per-suite settings. The baseline configurations and parameters of the optional extensions are not fully specified.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 4 and Appendix D report paired gains, sample counts, and 95% confidence intervals for the main controlled comparisons. Appendix D.1 defines the instance-cluster bootstrap with 10,000 resamples and all seeds retained within each cluster.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix D.2 identifies the NVIDIA GH200 and reports synchronized component times and runtime-matching ratios for the paired controls. Memory requirements and the total project compute budget are not fully specified.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work studies combinatorial optimization on public benchmark graphs and involves no human-subject experiments or release of a model trained on personal data.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 6 discusses potential uses in scheduling and allocation, and the risks of optimizing objectives that omit relevant social constraints.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No pretrained model or high-risk dataset is released with this manuscript.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The relevant algorithms and benchmark sources are cited. Complete version identifiers and license information for every underlying asset are not provided.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No separate dataset, pretrained model, or software package is released with this manuscript. The algorithm is specified in Algorithm 1.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or human-subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve human-subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: No LLM is a component of the optimization method or its guarantees. AI assistance with the experimental workflow, manuscript preparation, and schematic figure is disclosed at the end of Appendix D.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.