Ehsan kahrizi

Question 1 Combinatorial Problem

1. Introduction

In this problem, I investigated the performance of the Simulated Annealing (SA) algorithm on the Quadratic Assignment Problem (QAP). The objective of the optimization is to minimize the total flow cost between departments, which the cost is defined as the product of the flow between pairs of departments and the rectilinear distance between their assigned locations. The goal is determine an assignment of 15 departments to 15 locations. Also, I then analyzed how different algorithmic design choices affect the convergence behavior and solution quality. Specifically, three annealing schedules, three stopping criteria, and five different initial starting points were examined to evaluate the robustness and sensitivity of the SA algorithm on this combinatorial problem (Sonuc et al., 2018).

2. Methodology, Definitions and Assumptions

Problem Encoding. In this problem, the encoding of the problem is position of each department. Each position in the permutation represents a location, and the value stored at that position represents the department assigned to that location. For example:

$$\pi = [4,2,7,1,\dots,15]$$

means:

- Location 1 → Department 4
- Location 2 → Department 2
- Location 3 → Department 7

The objective function is defined as:

$$\text{Cost}(\pi) = \sum_{i=1}^{15}\sum_{j=1}^{15} F_{ij} \cdot D_{\pi(i)\pi(j)}$$

where: F_{ij} represents the flow between departments i and j; and D_{kl} denotes the distance between locations k and l. The goal is to minimize this total interaction cost.

Neighborhood Definition. The neighborhood of a solution is defined as all permutations that can be obtained by swapping two departments.

Move Operator. The move operator is a random pairwise swap: 1) Randomly select two distinct indices $i$and j; 2) Swap the departments assigned to those two locations.

Acceptance criterion. I used conditional acceptance criterion to determine whether a candidate solution replaces the current soloution, If the candidate produces a lower objective value, it is accepted deterministically. If the candidate results in a worse objective value, it is accepted probabilistically according to $e^{(-\Delta/T)}$, where Δ represents the increase in objective value and T is the current temperature.

Annealing Schedule and Initial Temperature. A geometric cooling schedule was selected for its simplicity and widespread use in practice. The temperature is updated according to:

$$T_{k+1} = \alpha T_k$$

where $\alpha \in (0,1)$ controls the cooling rate. Three values of α were tested (0.90, 0.95, and 0.98) to evaluate the influence of cooling speed. The initial temperature, T_0, was set as a fixed baseline to ensure a fair comparison across experiments.

Stopping criterion. The stopping mechanism in the implemented code is based on either a temperature threshold ($T_{\min} = 10^{-3}$ and $T_{\min} = 10^{-6}$) or a maximum number of function evaluations. In the baseline configuration, the algorithm terminates when the temperature falls below a predefined minimum value $T_{\min} = 10^{-3}$.

Experimental Design. To evaluate performanc and sensitivity, three annealing schedules, three stopping criteria, and five different initial starting points were tested, resulting in a total of 45 independent runs. Each run used a distinct random seed to maintain stochastic diversity while ensuring reproducibility. Performance metrics recorded include the best objective value found, the number of function evaluations, final temperature, and convergence trades.

3. Results and Discussion

3.1. Bassline configuration

In the baseline experiment, the SA algorithm was applied using a single cooling schedule and a random initial permutation. The best assignment found is shown *Figure 1*, where each department is mapped to a specific location. The *Figure 2* indicates a rapid reduction in cost during the early iterations, followed by a slower improvement phase. The best cost decreases sharply from the initial high value and stabilizes around the final solution, which showing that the algorithm was able to escape poor initial configurations and progressively refine the layout.

The *Figure 3* demonstrates a smooth exponential cooling process, where the temperature decreases steadily toward zero. As the temperature drops, the acceptance of worse solutions becomes less likely, leading to stabilization of the solution. The convergence behavior suggests that most meaningful improvements occurred in the mid-temperature range, while the final low-temperature phase mainly fine-tuned the assignment. Overall, the baseline results confirm that Simulated Annealing is capable of producing high-quality solutions for the QAP, although the final cost still depends on parameter settings and random initialization.

3.2. Annealing Schedule Comparison

The comparison of the three cooling schedules (α = 0.90, 0.95, 0.98) clearly shows the influence of cooling speed on solution quality (*Figure 4*). The fastschedule (α = 0.90) converges very quickly but becomes trapped in a poorer local minimum, that resulting in the highest final cost among the three. The moderate schedule (α = 0.95) improves the solution further and reaches a lower final cost. The slow cooling schedule (α = 0.98) maintains a higher temperature for a longer period, allowing more exploration and ultimately achieving the best solution (cost ≈ 599).

From the *Figure 6*, we observe that slower cooling leads to a more gradual and stable reduction in cost. Although α = 0.98 requires more evaluations to converge, it provides better escape capability from local minima and achieves superior final performance. This confirms the expected behavior of simulated annealing: slower cooling improves solution quality at the expense of computational effort, highlighting the exploration–exploitation tradeoff inherent in the algorithm.

3.3. Effect of Stopping Criteria

The comparison of stopping criteria shows that all three configurations converged to the same best cost value of 587, which indicating that the search process was robust enough to reach the same high-quality solution regardless of how termination was defined (*Figure 7*). However, the number of function evaluations required to reach this solution differed significantly. The strict temperature-based stopping rule with a very small T_{min}(1e-6) allowed the algorithm to cool longer, but the improvement in solution quality was not better than the other criteria. In contrast, the fixed evaluation budget (30,000 evaluations) provided a clear and predictable computational limit while still achieving the same optimal cost.

The *Figure 9* demonstrate that most of the improvement occurs early in the search process, during the higher temperature phase when exploration is dominant. After a certain point, the best cost stabilizes and additional evaluations mainly refine the solution without substantial gains.

3.4. Effect of Initial Starting Points Across All Combinations

This section highlights how sensitive the QAP solution is to the choice of initial permutation, even when the annealing schedule and stopping rule are fixed. Across the 3×3 combinations (three cooling rates × three stopping criteria), the individual assignment *Figure 10* show noticeable variability in the final layouts obtained from different starting points. Although many runs converge toward similar structural patterns, some initial permutations lead to slightly worse local minima, confirming that the search landscape of the QAP is highly multimodal.

The *Figure 11* further clarifies this behavior: combinations with slower cooling (α = 0.98) and stricter stopping (T_{min} = 1e-6) exhibit lower median costs and tighter spread, indicating both better performance and higher robustness against poor initializations. In contrast, faster cooling (α = 0.90) combined with loose stopping criteria shows larger variance and higher mean cost, suggesting premature convergence. The *Figure 12* summarizes this interaction clearly: the best average performance occurs in the slow-cooling, deep-search regime, demonstrating that while initialization matters, its negative impact can be mitigated by a sufficiently exploratory annealing schedule.

4. Conclusion

In this problem, the SA algorithm was applied to the Quadratic Assignment Problem and systematically analyzed under different cooling schedules, stopping criteria, and initial starting points. The results show that algorithmic design choices strongly influence both convergence speed and solution quality. Faster cooling leads to rapid convergence but increases the risk of getting trapped in local minima, while slower cooling allows deeper exploration and consistently produces lower costs.

Among all configurations, the combination of a slow cooling schedule (α = 0.98) and a strict stopping criterion (T_{min} = 1e-6) achieved the best average performance and the most stable results across different initial permutations. Although the QAP is highly multimodal and sensitive to initialization, but careful tuning of annealing parameters significantly improves robustness for solving suche problems.

5. Python code

The Python codes used for this project are shared in GoogleColab, which is an online, free-access, and reproducible platform, as follows:

[Python code for Problem 1]

Ehsan kahrizi

Question 2 Continuous Problem

1. Introduction

In this problem, I investigated the performance of the SA algorithm on the six-hump camelback function, which is a multimodal benchmark problem in continuous optimization. The objective of optimization here is to minimize the function within bounded domains. Then I analyzed how different algorithmic designe choices could affect the convergence process and results. Specifically, three annealing schedules, three stopping criteria, and five different initial starting points were examined (Nugent et al. 1968).

2. Methodology, Definitions and Assumptions

The objective is to minimize $z(x, y)$ with bounds: $x \in [-3,3]$ and $y \in [-2,2]$, which function is:

$$z(x, y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2)y^2$$

To implement SA, we developed a reproducible Python workflow that enables comprehensive SA. The implementation was designed to explicitly address the basic components required in the assignment: problem encoding, neighborhood definition, move operator design, annealing schedule selection (including initial temperature), and stopping criteria. All experiments were conducted using the Python with NumPy package for numerical computations and Matplotlib for visualization. Also, the stochastic nature of the algorithm was controlled using fixed random seeds to ensure reproducibility across runs.

Problem Encoding. The optimization problem was encoded as a continuous two-dimensional decision vector $X = (x, y)$ where $x \in [-3,3]$ and $y \in [-2,2]$. The objective function is to minimize the Z value within the specified bounded domain.

Neighborhood Definition. The neighborhood of a current solution was defined as the set of points generated by small perturbations in both decision variables. Neighbors are generated by applying a small random perturbation to the current solution. The ε_x and ε_y are random perturbations. A neighboring solution is defined as:

$$x' = x + \varepsilon_x, y' = y + \varepsilon_y,$$

Move Operator. The move operator, which implicitly defines the neighborhood structure, was implemented as a Gaussian random perturbation applied independently to both decision variables. Specifically, new candidate solutions were generated using normally distributed noise scaled by a temperature-dependent step size. The step size decreases proportionally to the square root of the ratio T/T_0, which promotes exploration during early iterations and exploitation during later stages. After generating a candidate, feasibility was enforced by clipping values to ensure they remained within the problem bounds.

Additionally, the Acceptance criterion, Annealing Schedule, Initial Temperature, stopping criterion, and Experimental Design that I used in this problem were the same as those I used for problem 1.

Assumptions. The following assumptions during this project were made:

- The objective function is deterministic and continuous.

- Boundary constraints are strictly enforced via clipping.
- The geometric cooling schedule is sufficient for convergence analysis.
- All parameter comparisons are performed under consistent baseline conditions to ensure fair evaluation.
- Stochastic variability is controlled through fixed seeds per run.

3. Results and Discussion

The following results show the results for each section.

3.1. Bassline configuration

The baseline SA implementation successfully converged to a near-global minimum of the six-hump camelback function. As shown in the Figure 13, the algorithm rapidly improved the objective value within the first few thousand function evaluations and quickly reached a value close to the known global minimum (approximately −1.0316). After this early convergenc phase, the best objective value remained stable for the remainder of the run. This indicating that the algorithm successfully located and retained a high-quality solution. This behavior suggests that the selected cooling rate ($\alpha = 0.95$) provided a reasonable balance between exploration and exploitation.

The Figure 14 shows the geometric cooling schedule, where temperature decreases smoothly and monotonically from the initial value of 5.0 toward zero. The exponential decay reflects the geometric rule ($T_{k+1} = \alpha\ T_k$), ensuring gradual reduction in randomness during the search. At high temperatures, the algorithm explores broadly, while at lower temperatures it behaves more like a local search method, refining the solution around promising regions.

The Figure 15 further confirms this behavior. Initially, the acceptance rate is relatively high (around 0.75–0.80), which indicating significant exploration and frequent acceptance of uphill moves. As the temperature decreases, the acceptance rate gradually declines to approximately 0.65, reflecting a transition toward more selective acceptance and stronger exploitation. This steady reduction demonstrates that the algorithm effectively transitions from stochastic exploration to focused local refinement, which is characteristic of a properly functioning Simulated Annealing implementation.

3.2. Annealing Schedule Comparison

The effect of the cooling rate α was evaluated using three geometric schedules: 0.90 (fast cooling), 0.95 (moderate), and 0.98 (slow cooling). As shown in the Figure *16*, smaller values of α reduce the temperature more quickly, while larger values maintain higher temperatures for a longer duration, which allows extended exploration.

The Figure *17* reflects this behavior. The slow cooling schedule ($\alpha = 0.98$) maintains a higher acceptance rate throughout the run, that indicating stronger exploration. In contrast, the fast-cooling schedule ($\alpha = 0.90$) shows a faster decline in acceptance rate, which leads to earlier exploitation. Despite these differences, all three schedules converge to approximately the same near-global minimum. The moderate schedule ($\alpha = 0.95$) provides a balanced trade-off between convergence speed and exploration stability.

3.3. Effect of Stopping Criteria

The impact of three stopping criteria was evaluated. In the case 1 and 3 the termination was set on the $T_{\min} = 10^{-3}$ and $T_{\min} = 10^{-6}$, while the evaluation budget was fixed on the 100000. In the case2 the evaluation budget (30,000 evaluations) was changed. As shown in the Figure *18*, the temperature-based stopping rules allow the cooling schedule to continue until the threshold is reached, while the evaluation-budget case stops earlier, regardless of the temperature level.

The *Figure 19* shows acceptance rate behavior is similar across all cases since the cooling schedule remains unchanged; however, the budget-based stopping rule truncates the search earlier, limiting further refinement. The stricter temperature threshold ($T_{\min} = 10^{-6}$) allows the algorithm to continue longer in the low-temperature regime, enabling slightly deeper exploitation. Despite, these differences in termination time, all three stopping criteria converge to approximately the same near-global minimum. This indicates that, for this problem, solution quality is relatively robust to the choice of stopping rule, although stricter stopping conditions require greater computational effort.

3.4. Effect of Initial Starting Points Across All Combinations

To evaluate robustness of the this optimization, five different starting points were tested for each combination of annealing schedule and stopping criterion, which resulting in 45 total runs. The Figure *20*, boxplot of best objective values across all 3×3 configurations shows that the majority of runs converge very close to the known global minimum. Alos, The spread of results is generally small, which indicating that the SA implementation is relatively insensitive to the choice of starting point.

However, some combinations exhibit slightly larger variability. In particular, configurations using the evaluation-budget stopping criterion (30k evaluations) show greater dispersion in final objective values, especially when combined with slower cooling schedules. This tell us, that early termination can limit the algorithms ability to fully refine the solution when exploration is prolonged. In contrast, temperature-based stopping rules ($T_{\min} = 10^{-3}$ and 10^{-6}) produce more consistent results across different initial conditions.

The Figure *20* of mean best objective values further confirms that overall solution quaility remains very close across all configurations, with only minor differences between combinations. This indicates, that the algorithm is good to initial conditions for this problem, which provided that sufficient cooling and search time are allowed. Overall, while starting points introduce some variability, their influence is significantly moderated by the cooling schedule and stopping criterion, and the algorithm reliably converges near the global minimum across most configurations. The Figure *21* shows the best combinations of annealing schedule and stopping criterial.

4. Conclusion

This problem evaluated the performance of SA algorithm on the six-hump camelback function under varying cooling schedules, stopping criteria, and initial starting points. The baseline implementation successfully converged to the known near-global minimum, demonstrating that the chosen encoding, neighborhood structure, and geometric cooling schedule were appropriate for this continuous multi-modal problem.

The comparison of annealing schedules showed that slower cooling rates maintain higher acceptance rates and longer exploration, while faster cooling accelerates convergence but increases the risk of premature exploitation. A moderate cooling rate ($\alpha = 0.95$) provided a balanced trade-off between exploration and convergence speed. Analysis of stopping criteria indicated that temperature-based stopping rules produce more consistent results, while strict evaluation budgets may slightly limit solution refinement. Finally, testing five different starting points across all 3×3 parameter combinations demonstrated that the algorithm is generally robust to initialization, with only minor variability observed under early stopping conditions.

5. Python code

The Python codes used for this project are shared in GoogleColab, which is an online, free-access, and reproducible platform, as follows:

Python code for Problem 2

References

Sonuc, E., Sen, B., & Bayir, S. (2018). A cooperative GPU-based parallel multistart simulated annealing algorithm for quadratic assignment problem. *Engineering science and technology, an international journal*, *21*(5), 843-849.

Nugent, C. E., Vollmann, T. E., & Ruml, J. (1968). An experimental comparison of techniques for the assignment of facilities to locations. *Operations research*, *16*(1), 150-173.

Ehsan kahrizi

Appendix A_Combinatorial problem: Baseline Configuration plots

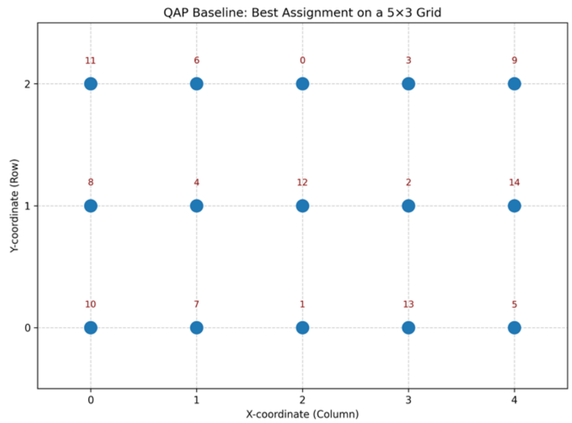

Figure 1 Best position map

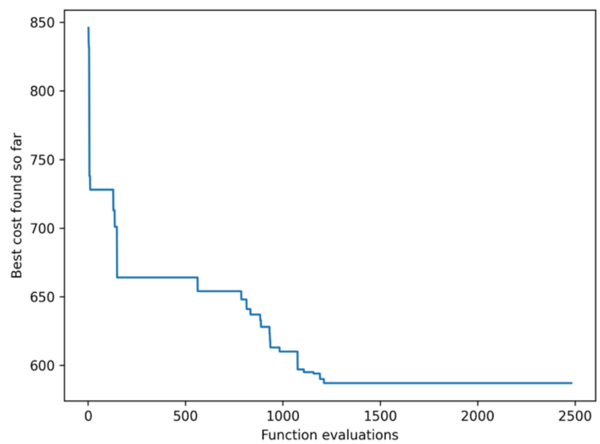

Figure 2 Best Cost vs Evaluations

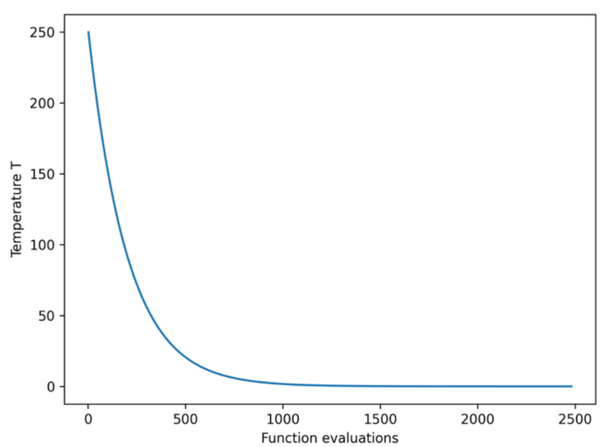

Figure 3 Temperature T vs evaluations

Ehsan kahrizi

Appendix B_Combinatorial problem: Annealing Schedule Comparison

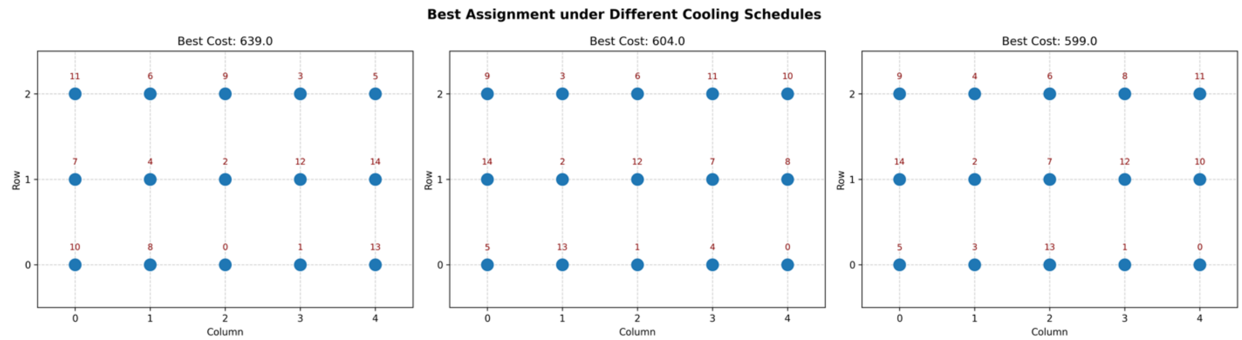

Figure 4 Best Position Map within different alpha

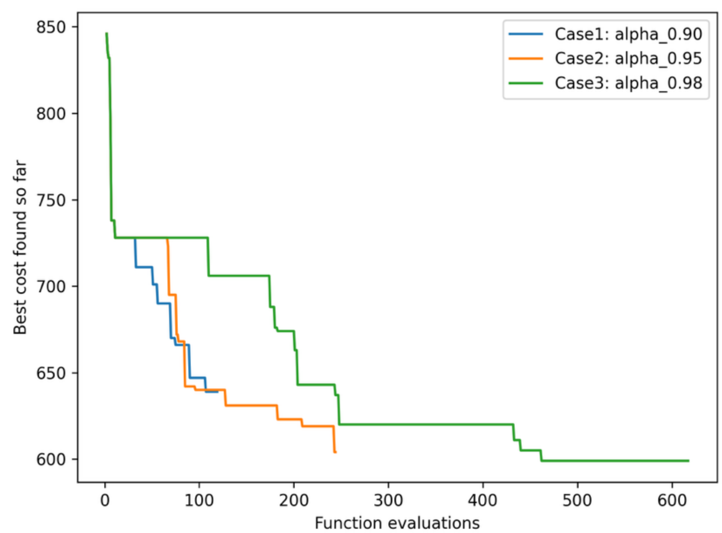

Figure 5 Best Cost vs Evaluations within different alpha

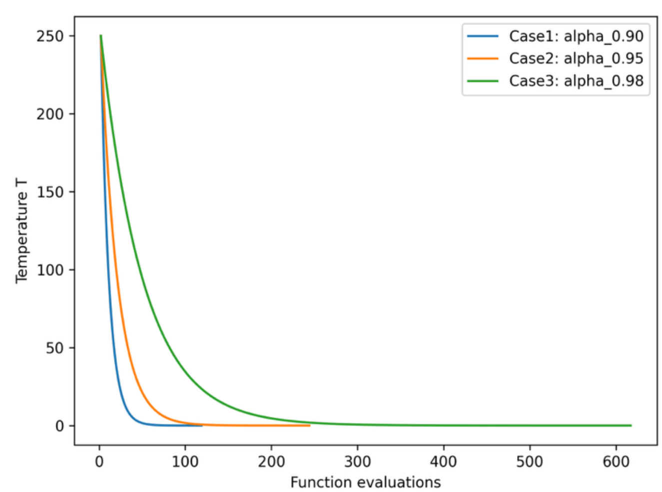

Figure 6 Temperature T vs evaluations within different alpha

Ehsan kahrizi

Appendix C_Combinatorial problem: Effect of Stopping Criteria

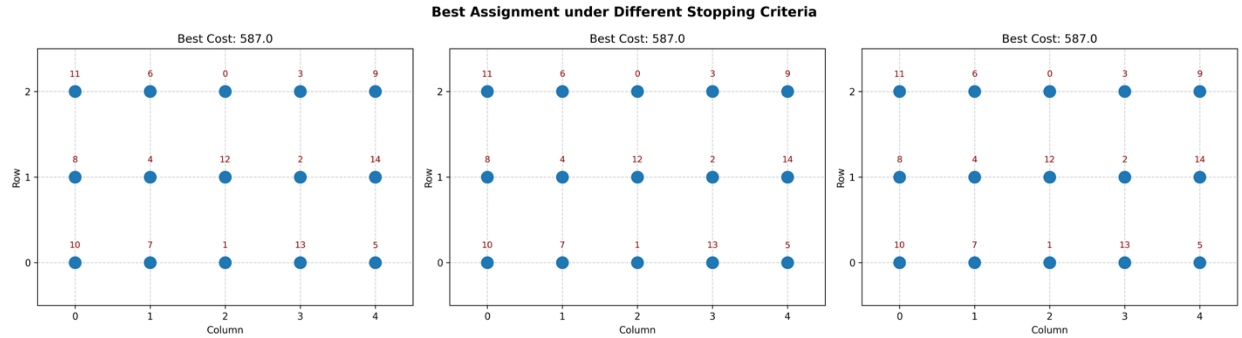

Figure 7 Best Position Map within different stop criteria

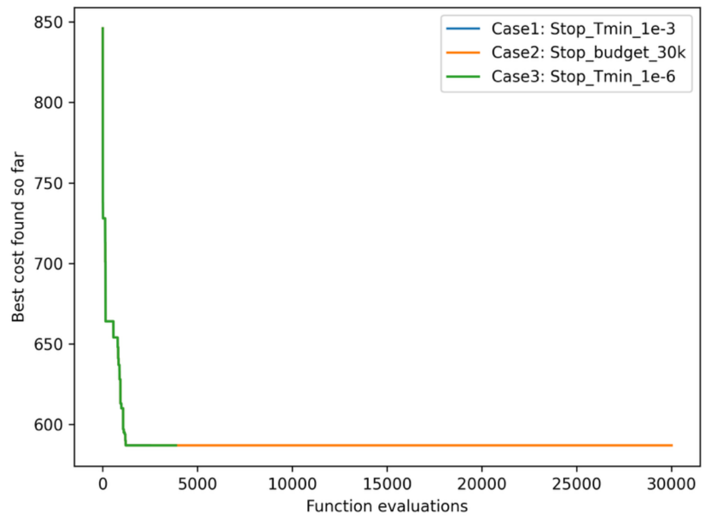

Figure 8 Best Cost vs Evaluations within different stop criteria

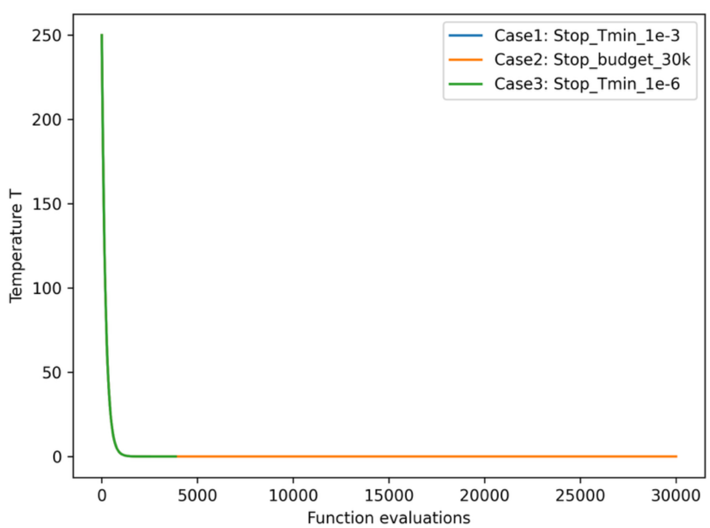

Figure 9 Temperature T vs evaluations within different stop criteria

Appendix D_Combinatorial problem: Effect of Initial Starting Points Across All Combinations

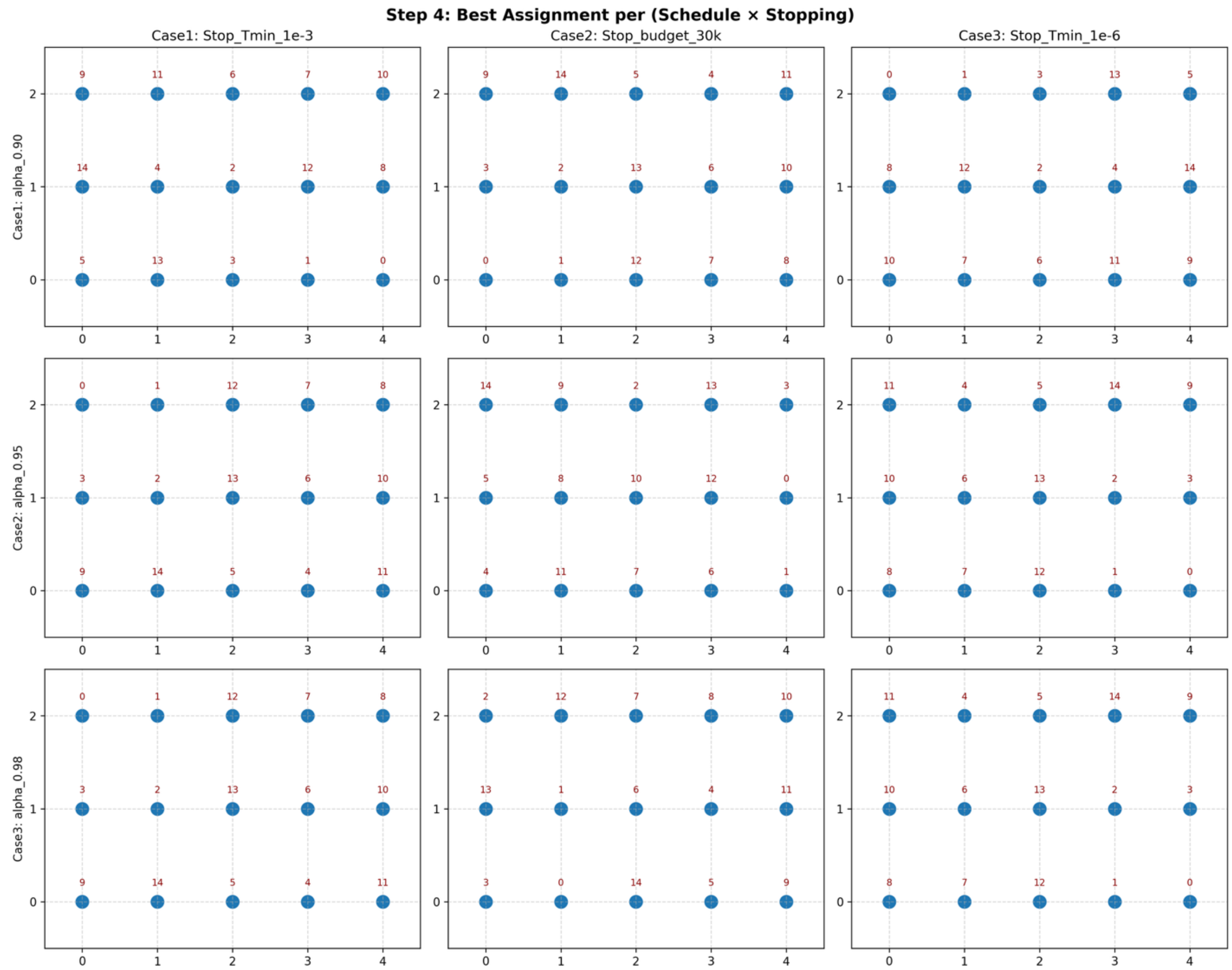

Figure 10 Best Position Map within different alpha and stop criteria

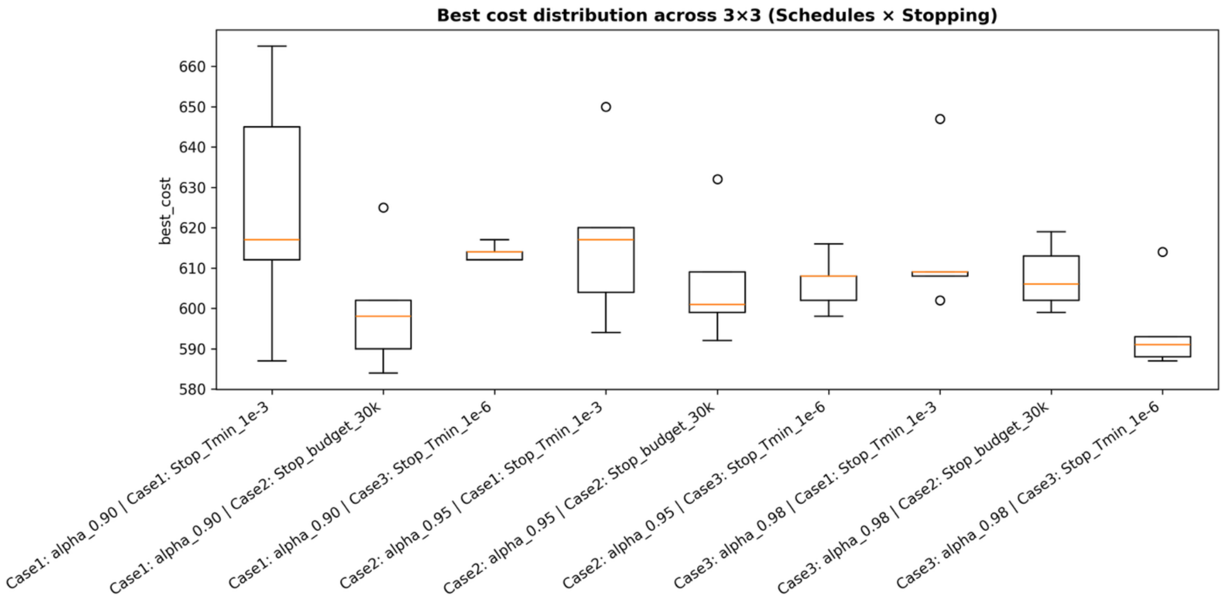

Figure 11 Best Cost distribution across 3×3 (Schedules × Stopping)

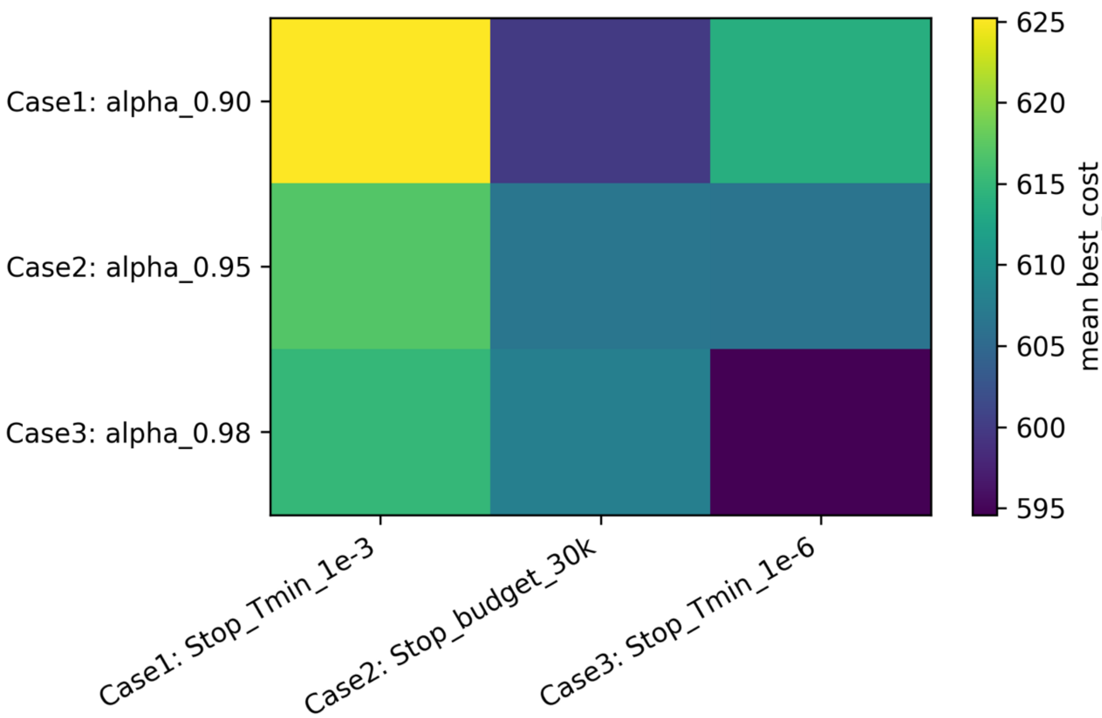

Figure 12 Mean best cost heatmap (schedule × stopping)

Appendix E_Continuous problem: Baseline Configuration plots

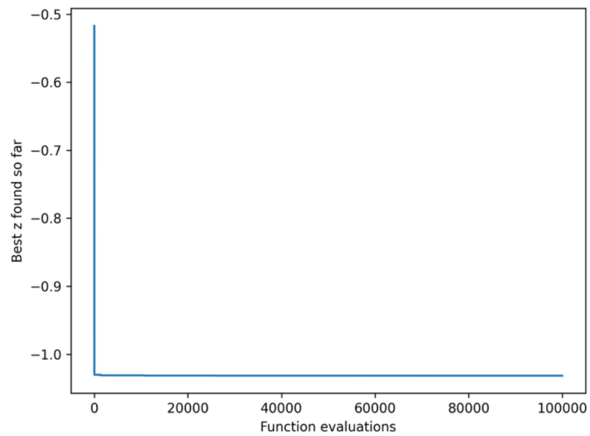

Figure 13 SA Baseline: Best z vs Evaluations

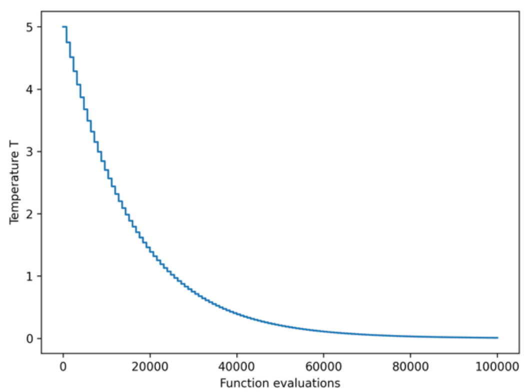

Figure 14 SA Baseline: Temperature vs Evaluations

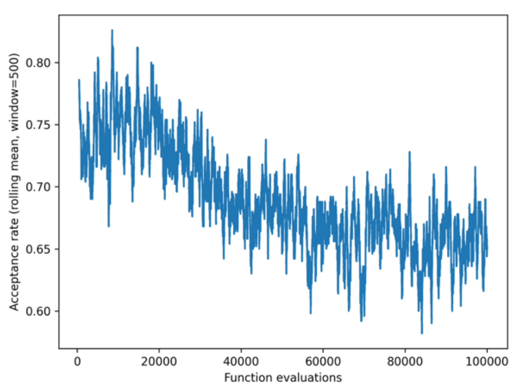

Figure 15 SA Baseline: Acceptance Rate

Table 1 Summary of the optimized solution

best_x	best_y	best_z	evals	T_0	alpha	T_{min}	seed	x_0	y_0
-0.0890	0.7128	-1.0316	100000	5	0.95	0.001	42	0	0

Appendix F_Continuous problem: Annealing Schedule Comparison

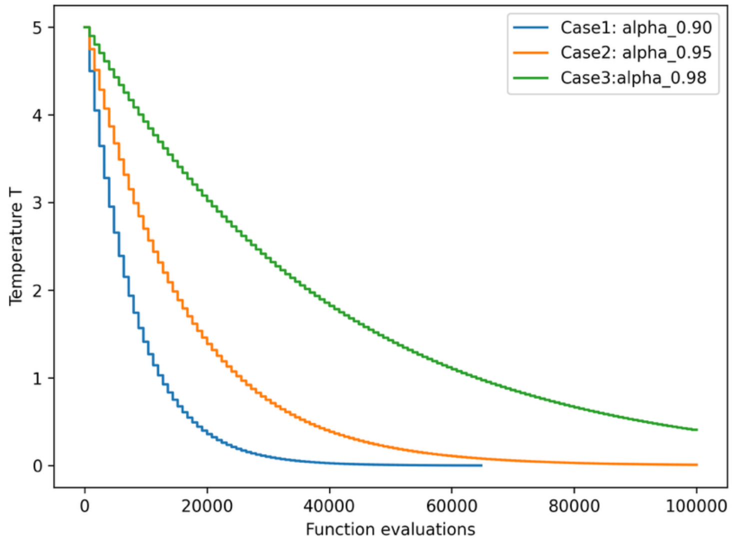

Figure 16 Annealing Schedule Comparison

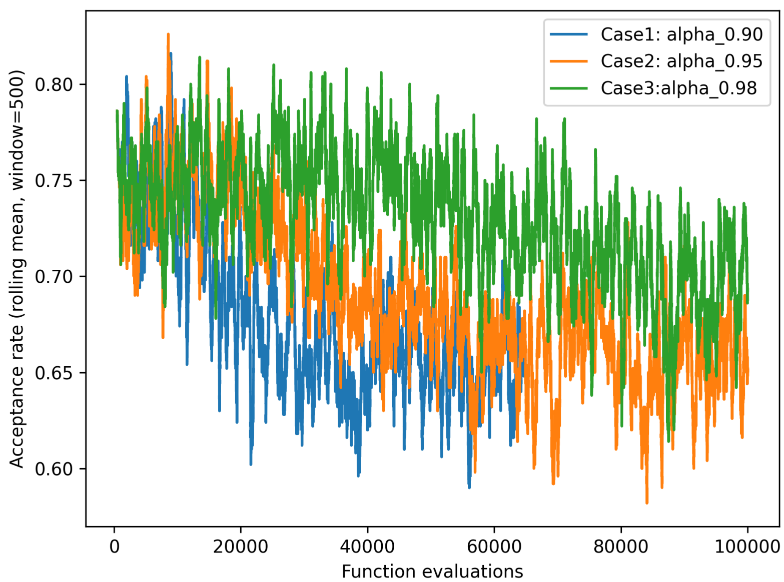

Figure 17 Annealing Schedule Comparison for different α

Table 2 Summary of the different cases for alpha

schedule	alpha	best_x	best_y	best_z	evals	final_T	T0	steps_per_T	Tmin	max_evals
Case1:	0.9	-0.0897	0.7127	-1.0316	64801	0.0010	5	800	0.001	100000
Case2:	0.95	-0.0890	0.7128	-1.0316	100000	0.0082	5	800	0.001	100000
Case3:	0.98	0.0899	-0.7133	-1.0316	100000	0.4002	5	800	0.001	100000

Appendix G_Continuous problem: Effect of Stopping Criteria

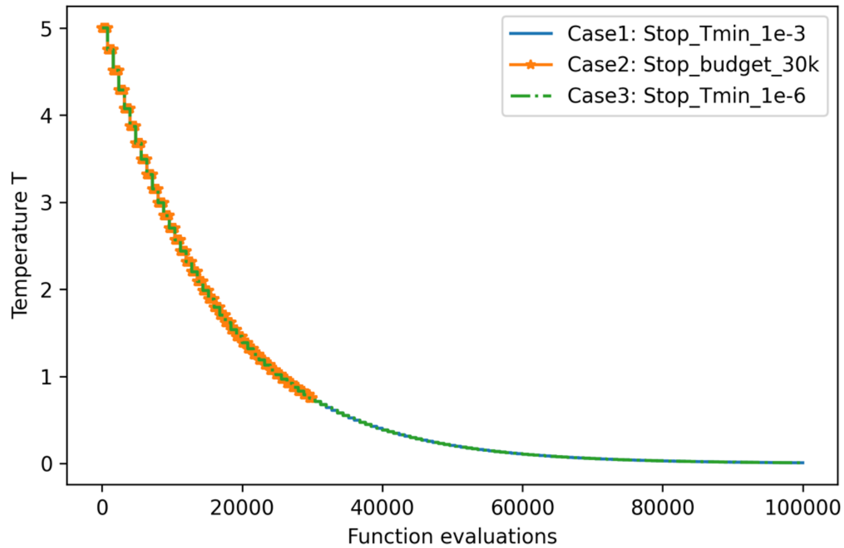

Figure 18 Temperature vs Evaluations (Different Stopping Criteria)

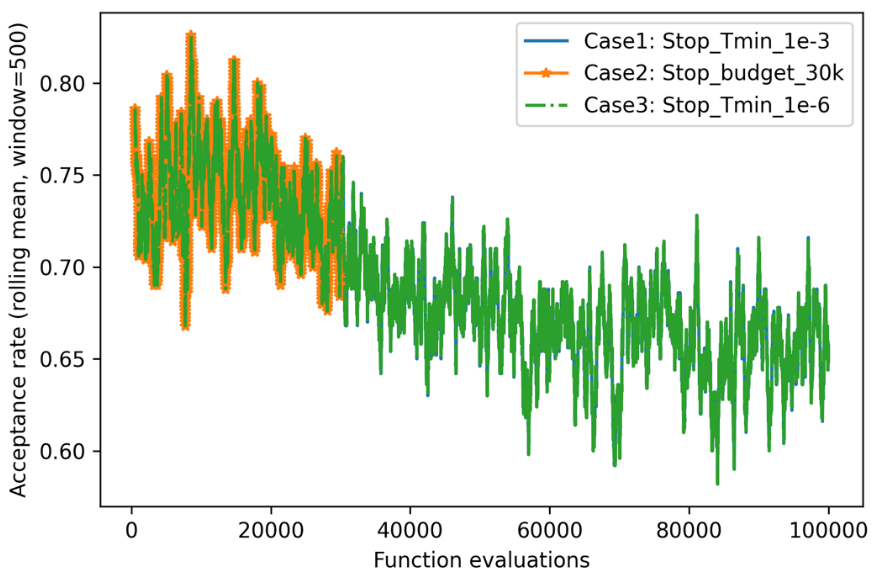

Figure 19 Acceptance Rate vs Evaluations

Table 3 Summary of the different cases for stopping

stop_name	Tmin	max_evals	best_x	best_y	best_z	evals
Case1: Stop_Tmin_1e-3	0.001	100000	-0.0890	0.7128	-1.0316	100000
Case2: Stop_budget_30k	0	30000	-0.0916	0.7101	-1.0316	30000
Case3: Stop_Tmin_1e-6	1.00E-06	100000	-0.0890	0.7128	-1.0316	100000

Appendix H_Continuous problem: Effect of Initial Starting Points Across All Combinations

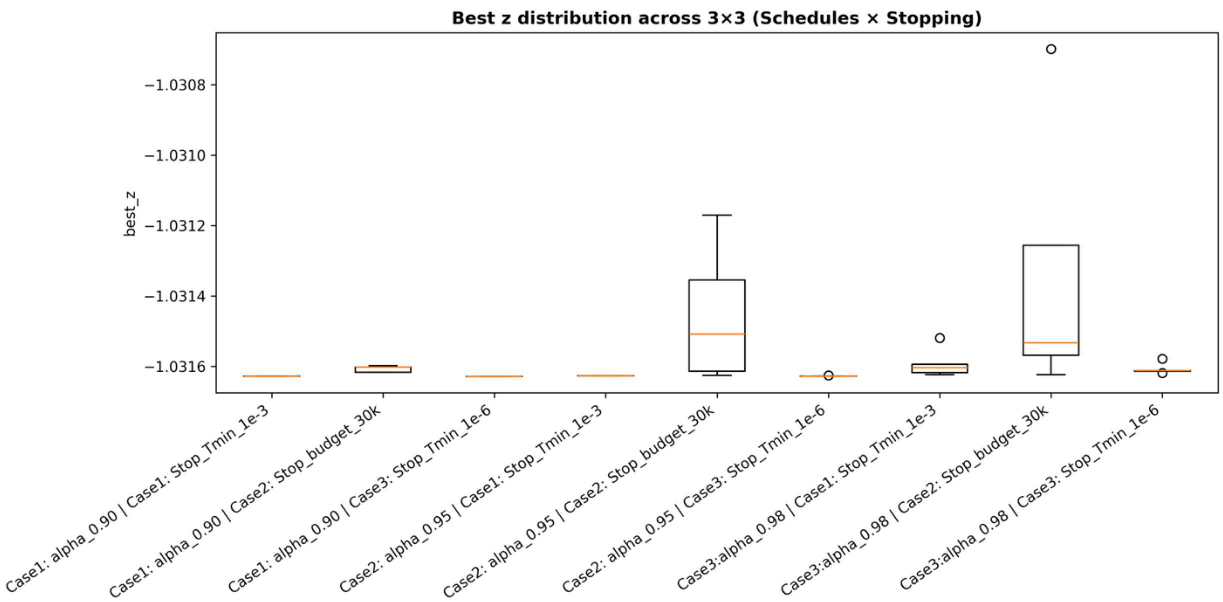

Figure 20 Best z distribution across 3×3 (Schedules × Stopping)

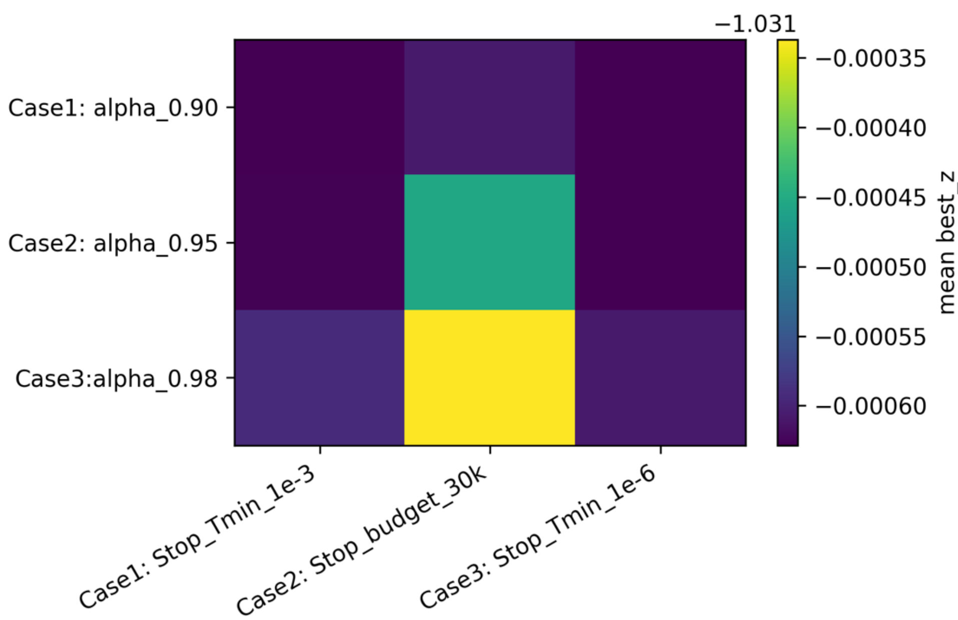

Figure 21 Mean best_z heatmap (schedule × stopping)

Table 4 Summary of 45 runs

run_id	schedule	stopping	T_{min}	max_evals	start	x_0	y_0	$best_x$	$best_y$	$best_z$	evals
1	Case1: alpha_0.90	Case1: Stop_Tmin_1e-3	0.001	100000	Case1: Start_(0,0)	0	0	0.0896	-0.7126	-1.0316	64801
2	Case1: alpha_0.90	Case1: Stop_Tmin_1e-3	0.001	100000	Case2: Start_(-3,-2)	-3	-2	0.0897	-0.7126	-1.0316	64801
3	Case1: alpha_0.90	Case1: Stop_Tmin_1e-3	0.001	100000	Case3: Start_(-3,2)	-3	2	-0.0896	0.7126	-1.0316	64801
4	Case1: alpha_0.90	Case1: Stop_Tmin_1e-3	0.001	100000	Case4: Start_(3,-2)	3	-2	-0.0901	0.7126	-1.0316	64801
5	Case1: alpha_0.90	Case1: Stop_Tmin_1e-3	0.001	100000	Case5: Start_(3,2)	3	2	0.0901	-0.7126	-1.0316	64801
6	Case1: alpha_0.90	Case2: Stop_budget_30 k	0	30000	Case1: Start_(0,0)	0	0	0.0884	-0.7132	-1.0316	30000
7	Case1: alpha_0.90	Case2: Stop_budget_30 k	0	30000	Case2: Start_(-3,-2)	-3	-2	-0.0905	0.7146	-1.0316	30000
8	Case1: alpha_0.90	Case2: Stop_budget_30 k	0	30000	Case3: Start_(-3,2)	-3	2	-0.0921	0.7119	-1.0316	30000
9	Case1: alpha_0.90	Case2: Stop_budget_30 k	0	30000	Case4: Start_(3,-2)	3	-2	0.0923	-0.7122	-1.0316	30000
10	Case1: alpha_0.90	Case2: Stop_budget_30 k	0	30000	Case5: Start_(3,2)	3	2	0.0885	-0.7119	-1.0316	30000
11	Case1: alpha_0.90	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case1: Start_(0,0)	0	0	-0.0898	0.7127	-1.0316	100000
12	Case1: alpha_0.90	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case2: Start_(-3,-2)	-3	-2	-0.0898	0.7127	-1.0316	100000
13	Case1: alpha_0.90	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case3: Start_(-3,2)	-3	2	0.0898	-0.7126	-1.0316	100000
14	Case1: alpha_0.90	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case4: Start_(3,-2)	3	-2	-0.0899	0.7127	-1.0316	100000
15	Case1: alpha_0.90	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case5: Start_(3,2)	3	2	0.0898	-0.7126	-1.0316	100000
16	Case2: alpha_0.95	Case1: Stop_Tmin_1e-3	0.001	100000	Case1: Start_(0,0)	0	0	0.0901	-0.7123	-1.0316	100000
17	Case2: alpha_0.95	Case1: Stop_Tmin_1e-3	0.001	100000	Case2: Start_(-3,-2)	-3	-2	0.0896	-0.7123	-1.0316	100000
18	Case2: alpha_0.95	Case1: Stop_Tmin_1e-3	0.001	100000	Case3: Start_(-3,2)	-3	2	-0.0892	0.7128	-1.0316	100000
19	Case2: alpha_0.95	Case1: Stop_Tmin_1e-3	0.001	100000	Case4: Start_(3,-2)	3	-2	0.0895	-0.7129	-1.0316	100000
20	Case2: alpha_0.95	Case1: Stop_Tmin_1e-3	0.001	100000	Case5: Start_(3,2)	3	2	-0.0892	0.7125	-1.0316	100000
21	Case2: alpha_0.95	Case2: Stop_budget_30 k	0	30000	Case1: Start_(0,0)	0	0	-0.0939	0.7103	-1.0315	30000
22	Case2: alpha_0.95	Case2: Stop_budget_30 k	0	30000	Case2: Start_(-3,-2)	-3	-2	0.0839	-0.7185	-1.0312	30000
23	Case2: alpha_0.95	Case2: Stop_budget_30 k	0	30000	Case3: Start_(-3,2)	-3	2	-0.0881	0.7130	-1.0316	30000
24	Case2: alpha_0.95	Case2: Stop_budget_30 k	0	30000	Case4: Start_(3,-2)	3	-2	0.0893	-0.7068	-1.0314	30000

25	Case2: alpha_0.95	Case2: Stop_budget_30k	0	30000	Case5: Start_(3,2)	3	2	-0.0904	0.7123	-1.0316	30000
26	Case2: alpha_0.95	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case1: Start_(0,0)	0	0	-0.0902	0.7128	-1.0316	100000
27	Case2: alpha_0.95	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case2: Start_(-3,-2)	-3	-2	-0.0904	0.7123	-1.0316	100000
28	Case2: alpha_0.95	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case3: Start_(-3,2)	-3	2	-0.0895	0.7125	-1.0316	100000
29	Case2: alpha_0.95	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case4: Start_(3,-2)	3	-2	-0.0896	0.7125	-1.0316	100000
30	Case2: alpha_0.95	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case5: Start_(3,2)	3	2	0.0901	-0.7128	-1.0316	100000
31	Case3:alpha_0.98	Case1: Stop_Tmin_1e-3	0.001	100000	Case1: Start_(0,0)	0	0	-0.0896	0.7147	-1.0316	100000
32	Case3:alpha_0.98	Case1: Stop_Tmin_1e-3	0.001	100000	Case2: Start_(-3,-2)	-3	-2	0.0884	-0.7120	-1.0316	100000
33	Case3:alpha_0.98	Case1: Stop_Tmin_1e-3	0.001	100000	Case3: Start_(-3,2)	-3	2	-0.0879	0.7137	-1.0316	100000
34	Case3:alpha_0.98	Case1: Stop_Tmin_1e-3	0.001	100000	Case4: Start_(3,-2)	3	-2	0.0944	-0.7148	-1.0315	100000
35	Case3:alpha_0.98	Case1: Stop_Tmin_1e-3	0.001	100000	Case5: Start_(3,2)	3	2	0.0903	-0.7120	-1.0316	100000
36	Case3:alpha_0.98	Case2: Stop_budget_30k	0	30000	Case1: Start_(0,0)	0	0	-0.0935	0.7119	-1.0316	30000
37	Case3:alpha_0.98	Case2: Stop_budget_30k	0	30000	Case2: Start_(-3,-2)	-3	-2	0.0847	-0.7066	-1.0313	30000
38	Case3:alpha_0.98	Case2: Stop_budget_30k	0	30000	Case3: Start_(-3,2)	-3	2	-0.0747	0.7096	-1.0307	30000
39	Case3:alpha_0.98	Case2: Stop_budget_30k	0	30000	Case4: Start_(3,-2)	3	-2	0.0899	-0.7134	-1.0316	30000
40	Case3:alpha_0.98	Case2: Stop_budget_30k	0	30000	Case5: Start_(3,2)	3	2	0.0850	-0.7116	-1.0315	30000
41	Case3:alpha_0.98	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case1: Start_(0,0)	0	0	-0.0887	0.7149	-1.0316	100000
42	Case3:alpha_0.98	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case2: Start_(-3,-2)	-3	-2	-0.0881	0.7131	-1.0316	100000
43	Case3:alpha_0.98	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case3: Start_(-3,2)	-3	2	0.0911	-0.7138	-1.0316	100000
44	Case3:alpha_0.98	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case4: Start_(3,-2)	3	-2	-0.0879	0.7122	-1.0316	100000
45	Case3:alpha_0.98	Case3: Stop_Tmin_1e-6	1.00E-06	100000	Case5: Start_(3,2)	3	2	0.0884	-0.7130	-1.0316	100000