CMDO: A COGNITIVE MEMORY-DRIVEN OPTIMIZATION ALGORITHM FOR ADAPTIVE POPULATION-BASED SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Population-based optimization methods often use previous search information through successful solutions, parameter adaptation, or operator performance, but they rarely retain the context in which a search behavior succeeded or failed. We introduce Cognitive Memory-Driven Optimization (CMDO), a derivative-free population-based optimizer that represents experience as the relationship between search context, search behavior, and observed outcome. CMDO organizes these experiences across working, episodic, and consolidated memory, retrieves them according to similarity with the current search state, and uses both positive and negative evidence to guide subsequent search. Retrieved experience does not replay previous candidate locations; instead, it selects search recipes that are reconstructed from the current population through exploratory, directed, and local search behaviors with adaptive search geometry. We evaluate CMDO on selected Blackbox Optimization Benchmarking test suite on COCO (BBOB/COCO) and Congress on Evolutionary Computation 2017 (CEC2017) problems against DE, CMA-ES, SHADE, GWO, HHO, and ORCA, and further study its application to seven-parameter photovoltaic model estimation using measured current–voltage data. The results show problem-dependent but competitive optimization performance, including the lowest median error among the compared methods on CEC2017 F10. More importantly, analysis of the search traces shows that context-dependent recall changes the distribution of executed search behaviors, while unsuccessful experiences remain available as negative evidence for later decisions, showing that accumulated experience directly shapes subsequent search behavior. These results support the use of explicit context–behavior–outcome memory as an active mechanism for controlling population-based search.

1 INTRODUCTION

Optimization methods must continually balance exploration of new regions with refinement of promising solutions. Existing approaches achieve this through population dynamics, stochastic search, parameter adaptation, or information retained from previous evaluations. Differential Evolution and CMA-ES adapt search through population differences and evolving sampling distributions (Storn & Price, 1997; Hansen et al., 2003), while methods such as Gray Wolf Optimization (GWO), Harris Hawk Optimization (HHO), and ORCA optimization use alternative population-based search dynamics (Mirjalili et al., 2014; Heidari et al., 2019; Golilarz et al., 2020). Other approaches explicitly use search history for example, SHADE adapts parameters from successful past values, and adaptive operator-selection methods use previous operator performance to guide future choices (Tanabe & Fukunaga, 2013; Durgut et al., 2021). These methods show the value of past information, but they also raise a more fundamental question: *what should an optimizer remember?*

A successful search step is meaningful not only because it improved a solution, but also because of the conditions under which it worked. The same behavior may be useful in one stage of optimization and ineffective in another. This suggests that optimization memory should capture more than a good solution, parameter value, or operator score. Instead, it can represent the relationship between the *current search state*, the *behavior applied in that state*, and the *outcome that followed*. Such a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

representation allows previous experience to be reused only when it is relevant to the present search condition.

This idea is closely related to the functional role of memory in neurocognitive systems, where memory supports adaptation through context-sensitive storage, retrieval, and reuse of experience (Golilarz et al., 2026). Related work in episodic control, memory-augmented neural models, and Hebbian memory has similarly shown the value of recalling prior experience according to the current context (Blundell et al., 2016; Pritzel et al., 2017; Rae et al., 2020; Wu et al., 2022; Money et al., 2026; Mujawar & Golilarz, 2026a;b). These ideas motivate a shift from treating memory as passive storage toward using it as an active mechanism for controlling future behavior.

We introduce *Cognitive Memory-Driven Optimization* (CMDO), an optimization framework in which search experience directly shapes future decisions. CMDO stores experiences that connect search context, search behavior, and observed outcome. These experiences are maintained across working, episodic, and consolidated memory, retrieved according to contextual similarity, and updated through reinforcement, consolidation, and forgetting. CMDO also retains negative experience, allowing previously unsuccessful behavior to reduce the likelihood of repeating similar search decisions under comparable conditions. Retrieved experience then influences how exploratory, directed, and local search behaviors are constructed. We evaluate CMDO on the BBOB/COCO benchmark suite (Hansen et al., 2021) and the CEC2017 single-objective bound-constrained benchmark suite (Awad et al., 2016), and on photovoltaic parameter estimation using a two-diode model and measured current–voltage data Muhammad et al. (2019). Comparisons include DE, CMA-ES, SHADE, GWO, HHO, and ORCA under matched experimental conditions. The experiments assess both optimization performance and the contribution of the proposed memory mechanisms.

The main contributions of this work are:

- We introduce CMDO, a memory-driven optimization framework that represents experience through the relationship between search context, applied behavior, and observed outcome.
- We develop a multi-timescale memory architecture with context-dependent retrieval, reinforcement, consolidation, forgetting, and continued acquisition of experience.
- We incorporate both positive and negative search experience so that useful behaviors can be reinforced while previously unsuccessful behaviors are discouraged in related search states.
- We evaluate CMDO across numerical benchmarks and photovoltaic parameter estimation.

The remainder of this paper is organized as follows. Section 2 reviews related work on adaptive optimization, episodic and context-dependent memory, and neurocognitive memory mechanisms. Section 3 presents CMDO, including its search context, search recipes, multi-timescale memory, contextual retrieval, and memory update process. Section 4 reports the numerical benchmark and photovoltaic parameter-estimation results and analyzes how recalled experience changes the search behavior, and discusses the implications and limitations of the proposed approach. Section 5 concludes the paper. Additional algorithmic, implementation, and experimental details are provided in the appendix.

2 Related Work and Motivation

2.1 Memory and adaptation in optimization

Optimization algorithms use past search information in different ways. Differential Evolution and CMA-ES adapt search through population relationships and evolving sampling distributions (Storn & Price, 1997; Hansen et al., 2003), while methods such as GWO, HHO, and ORCA define alternative population-based search dynamics (Mirjalili et al., 2014; Heidari et al., 2019; Golilarz et al., 2020). More explicit use of search history appears in adaptive evolutionary methods. SHADE stores successful control-parameter values and uses them to guide future parameter generation (Tanabe & Fukunaga, 2013), while adaptive operator-selection methods update operator preferences according to observed performance (Durgut et al., 2021). Related studies further show that adaptive parameter control can strongly influence Differential Evolution behavior (Tanabe, 2020; Tanabe & Fukunaga, 2020).

These approaches demonstrate the value of past performance, but CMDO focuses on a different question, *under what search condition did a particular behavior succeed or fail?* The usefulness of a search action may depend on population diversity, search progress, stagnation, and proximity to promising regions. CMDO therefore links search behavior to the state in which it was applied, rather than treating success as independent of context.

2.2 Episodic and context-dependent memory

Context-dependent reuse of experience has been studied extensively outside numerical optimization. Model-Free Episodic Control and Neural Episodic Control store prior experiences and retrieve related states to support later decisions (Blundell et al., 2016; Pritzel et al., 2017). Experience replay similarly reuses previous interactions during learning, with prioritized replay emphasizing experiences expected to provide stronger learning signals (Schaul et al., 2016). Generalizable episodic memory and episodic curiosity further show how stored experience can be compared with current representations to influence future behavior (Hu et al., 2021; Savinov et al., 2019).

Memory-augmented sequence models provide a complementary perspective. Compressive Transformers, Memorizing Transformers, and Recurrent Memory Transformers maintain information beyond the immediate processing window (Rae et al., 2020; Wu et al., 2022; Bulatov et al., 2022). Although these methods address different tasks, they support a common principle relevant to CMDO and that is, memory is most useful when the system determines what should be retained, when it should be retrieved, and how it should affect current computation.

2.3 From neurocognitive memory to search experience

Neurocognitive-inspired intelligence treats memory as an active component of adaptation rather than passive storage. The framework in Golilarz et al. (2026) describes working memory as a rapidly accessible store for current information and longer-term memory as a mechanism for retaining context-rich and consolidated experience. Retrieval is associative and sensitive to context, while consolidation allows selected experience to persist and guide future behavior. Recent Hebbian memory models provide related computational examples. Hebbian fast weights form temporary associative memories during an episode (Money et al., 2026), adaptive Hebbian routing regulates memory contribution, plasticity, and retention according to the current task (Mujawar & Golilarz, 2026a), and Hierarchical Hebbian Memory organizes experience across working, episodic, and more stable memory levels (Mujawar & Golilarz, 2026b).

CMDO transfers these functional ideas to optimization, but uses a different memory object. Rather than storing only neural representations, candidate solutions, successful parameters, or aggregate operator rewards, CMDO stores a *search experience*, the search state, the behavior applied in that state, and the resulting outcome. These experiences are maintained across working, episodic, and consolidated memory and are retrieved according to similarity with the current search context. CMDO also retains negative evidence. A behavior that failed under a similar search condition can reduce support for repeating that behavior, while successful experience can strengthen it. Thus, memory is used not only to recall what worked, but also to avoid repeating previously unproductive search behavior.

The resulting distinction is central to CMDO, rather than adapting parameters independently, selecting operators from aggregate reward, or recalling earlier solutions, CMDO retrieves context-conditioned experience describing *what was tried, under what search condition, and what consequence followed*. The next section formalizes how these experiences are represented, stored, retrieved, and converted into new optimization actions.

3 Cognitive Memory-Driven Optimization Algorithm

3.1 Overview

We consider bounded continuous optimization,

$$\min_{\mathbf{x}\in\Omega} f(\mathbf{x}), \tag{1}$$

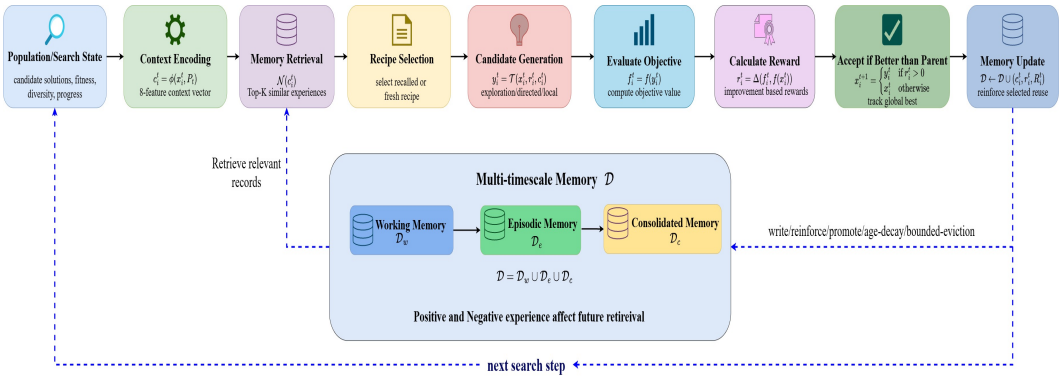


Figure 1: Overall architecture of Cognitive Memory-Driven Optimization (CMDO).

where $\Omega \subset \mathbb{R}^D$ is the search space. CMDO maintains a population of candidate solutions and generates one new candidate at each search step.

The main difference in CMDO is how previous search information is represented. Instead of storing only a good solution or a successful parameter value, CMDO stores a complete search experience,

$$\mathcal{E} = (\mathbf{c}, \mathbf{r}, R)\,, \tag{2}$$

where $\mathbf{c}$ is the current search context, $\mathbf{r}$ is the search recipe used in that context, and R is the resulting reward. When a similar search condition appears later, CMDO retrieves relevant past experiences and uses them to guide the next search action. The overall CMDO architecture is illustrated in Figure 1. The search process forms a closed loop in which the current population determines the search context, memory influences the next search behavior, and the observed outcome is written back as new experience for future retrieval.

3.2 Search Context

The search context summarizes the current condition of the population. CMDO uses an eight-dimensional context vector,

$$\mathbf{c} = [c_1, c_2, \ldots, c_8] \tag{3}$$

The eight components describe: (1) population spread, (2) the rank of the focal solution, (3) its distance to the current best solution, (4) its distance to the population centroid, (5) the fraction of the evaluation budget consumed, (6) current stagnation, (7) the recent frequency of positive rewards, and (8) the fitness gap relative to the current best solution. These features allow CMDO to distinguish between different search situations. The context uses only information available during optimization; knowledge of the true optimum is never used by the search controller.

3.3 Search Recipe and Candidate Generation

A search recipe defines how a new candidate should be generated:

$$\mathbf{r} = (o, s, m, g), \tag{4}$$

where o is the search operator, s is the step scale, m controls the directional mixture, and g determines whether the search acts on the full space or on a subset of dimensions. CMDO uses three search behaviors which are, *exploration*, *directed search*, and *local refinement*. Let $\mathbf{x}_i$ be the focal solution, $\mathbf{x}_{best}$ the current best solution, and $\boldsymbol{\delta}$ a difference vector obtained from two population members. Let $\boldsymbol{\epsilon}$ denote a population-scaled random perturbation. The three search directions are

$$\mathbf{d}_{exp} = m\boldsymbol{\delta} + (1-m)\boldsymbol{\epsilon}, \tag{5}$$

$$\mathbf{d}_{dir} = m(\mathbf{x}_{best} - \mathbf{x}_i) + (1-m)\boldsymbol{\delta}, \tag{6}$$

$$\mathbf{d}_{loc} = 0.1\,[m(\mathbf{x}_{best} - \mathbf{x}_i) + (1-m)\boldsymbol{\epsilon}] \tag{7}$$

A new candidate is generated as

$$\mathbf{x}' = \mathbf{x}_i + s\mathbf{d}. \tag{8}$$

The recipe therefore controls both the type and strength of the search move. The geometry component can restrict the move to a randomly selected subset of approximately $\sqrt{D}$ dimensions, allowing CMDO to alternate between full-space and lower-dimensional search.

3.4 Multi-timescale memory

CMDO organizes search experience into three bounded memory levels, *working memory*, *episodic memory*, and *consolidated memory*. Working memory stores recent experiences, episodic memory retains a larger history over a longer period, and consolidated memory preserves patterns that have repeatedly produced useful outcomes. The three levels store the context, recipe, and outcome associated with search behavior rather than only candidate locations. This allows CMDO to recall *how to search* rather than simply *where a good solution was found*.

3.5 Context-dependent retrieval

For the current context $\mathbf{c}$ and a stored context $\mathbf{c}_j$, CMDO measures their similarity using

$$S_j = \exp\left(-\frac{\mathrm{MSE}(\mathbf{c}, \mathbf{c}_j)}{2h^2}\right), \tag{9}$$

where h controls how strongly context differences affect retrieval.

Older memories gradually lose influence through a retention factor,

$$W_j = S_j \rho^{a_j}, \tag{10}$$

where a_j is the age of the memory and ρ is the retention coefficient. CMDO keeps only sufficiently relevant memories and considers the most relevant records for recipe selection. A stored experience receives reuse credit only when its recipe is actually selected and executed.

3.6 Learning from positive and negative experience

CMDO learns from both successful (positive) and unsuccessful (negative) search behavior. Positive experience supports repeating a recipe when a related context appears again, while negative experience reduces support for similar behaviors that previously failed under comparable conditions.

The support for a retrieved recipe is

$$Q_j = \text{positive support} - \lambda\, \text{negative evidence}, \tag{11}$$

where λ controls the influence of failure information.

Negative evidence is applied only when the failed experience is behaviorally similar to the candidate recipe, including the same search operator and geometry. If no recalled recipe has sufficient support, CMDO generates a new recipe. A fixed probability of generating a fresh recipe is also retained so that new search behaviors can continue to be discovered.

3.7 Reward and memory update

After evaluating a candidate, CMDO compares its objective value with that of the focal solution. The reward is computed as

$$R = \tanh\left(\frac{f(\mathbf{x}_i) - f(\mathbf{x}')}{\sigma_f}\right), \tag{12}$$

where σ_f is a robust scale estimate from the current population fitness values.

A positive reward indicates improvement, while a non-positive reward records an unsuccessful action. The candidate replaces the focal solution only when it improves the objective. The resulting experience is stored in working and episodic memory, and repeatedly useful experience may later be consolidated. If the executed recipe was recalled, only the selected source receives reuse feedback.

The overall CMDO cycle can be summarized as

Table 1: Median final error over three seeds on BBOB and CEC2017 benchmarks (lower is better).

Method	BBOB ($D = 5$)			CEC2017 ($D = 10$)		
	F1	**F8**	**F15**	**F1**	**F4**	**F10**
CMDO	1.450×10^{-3}	5.7813	21.4916	3.7848×10^{9}	189.9318	1322.7172
HHO	2.406×10^{-1}	35.9889	47.1264	1.1906×10^{10}	566.7792	1622.6580
GWO	1.088×10^{-2}	3.9398	21.7654	3.8409×10^{8}	15.6910	1415.7922
ORCA	1.038×10^{-2}	3.2394	11.5687	1.2090×10^{10}	1900.3090	1586.3345
DE	5.859×10^{-4}	3.8711	20.5967	9.6074×10^{6}	7.6000	1946.1414
CMA-ES	9.861×10^{-3}	4.5501	31.7369	1.7650×10^{7}	8.4741	2070.8946
SHADE	1.263×10^{-1}	21.2723	15.0919	3.3633×10^{7}	16.4747	1800.6883

Observe → Recall/Explore → Act → Evaluate → Update → Remember

The cycle emphasizes that memory controls how search behavior is selected, while each recalled recipe is reconstructed from the current population rather than replaying a previously visited solution.

4 Experimental Results and Discussion

4.1 Numerical optimization performance

Table 1 reports the median final optimization error over seeds 15–17. Lower values indicate better performance. The two benchmark suites are presented separately because their objective scales differ substantially. On BBOB F1, CMDO achieves a median error of 1.45×10^{-3}, lower than HHO, GWO, ORCA, CMA-ES, and SHADE. On F8 it improves over HHO and SHADE, while on F15 it remains close to GWO and DE and improves over HHO and CMA-ES. The results indicate that the fixed CMDO configuration remains competitive across different BBOB landscapes, although its relative performance varies by problem.

On CEC2017 F1 and F4, CMDO improves over HHO and ORCA, while several evolutionary baselines obtain lower errors. On F10, CMDO reaches a median error of 1322.72, lower than all six comparison methods. This is the strongest numerical benchmark result for CMDO in the reported experiments. Figure 2a shows that CMDO continues to improve throughout the available evaluation budget on F10 and finishes with a lower median best-so-far error than HHO, GWO, and ORCA.

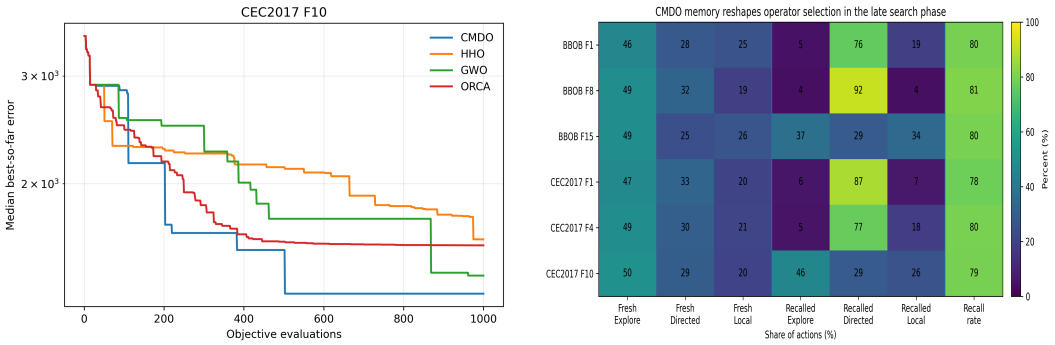
4.2 How memory changes the search

The main question behind CMDO is whether stored experience actually changes future search behavior. We examine this by separating newly generated *fresh* recipes from recipes selected through memory recall.

Figure 2b shows a clear difference between fresh and recalled behavior. Fresh recipes remain close to the predefined operator sampling distribution, whereas recalled recipes develop problem-dependent preferences. Directed search becomes dominant among recalled actions on BBOB F1, BBOB F8, CEC2017 F1, and CEC2017 F4, while BBOB F15 shows a more balanced mixture of exploration, directed search, and local refinement. CEC2017 F10 exhibits another pattern, with recalled actions maintaining a mixed operator distribution. Because the fresh-recipe distribution is fixed, these shifts arise from accumulated search experience rather than from a change in the underlying operator prior. Memory therefore does more than store previous evaluations: it changes which search behaviors are selected under related search conditions. The variation across functions also indicates that CMDO does not learn one globally preferred operator.

4.3 Negative experience and memory persistence

CMDO retains unsuccessful as well as successful search experiences. Negative records remain available during retrieval and can reduce support for similar recipes when related contexts are encoun-



(a) Convergence on CEC2017 F10 (median best-so-far error over seeds 15–17; lower is better).

(b) Memory-driven search behavior in the late optimization phase.

Figure 2: Comparison of convergence and memory-driven search behavior.

tered again. This is particularly visible on CEC2017 F10, where the late search memory contains substantially more negative than positive original experiences. These failures are therefore not discarded simply because they did not improve the focal solution. The three memory levels provide complementary time scales for this experience. Working memory retains recent events, episodic memory maintains a larger history of positive and negative outcomes, and consolidated memory preserves patterns that have demonstrated useful reuse. Together with age-dependent retention and bounded capacity, this allows recent evidence to remain responsive while repeatedly useful behavior can persist for longer. These observations support the central design of CMDO where, the optimizer remembers *what was tried, under what search condition, and what happened*, using both successful and unsuccessful experience to influence later behavior.

4.4 Photovoltaic parameter estimation

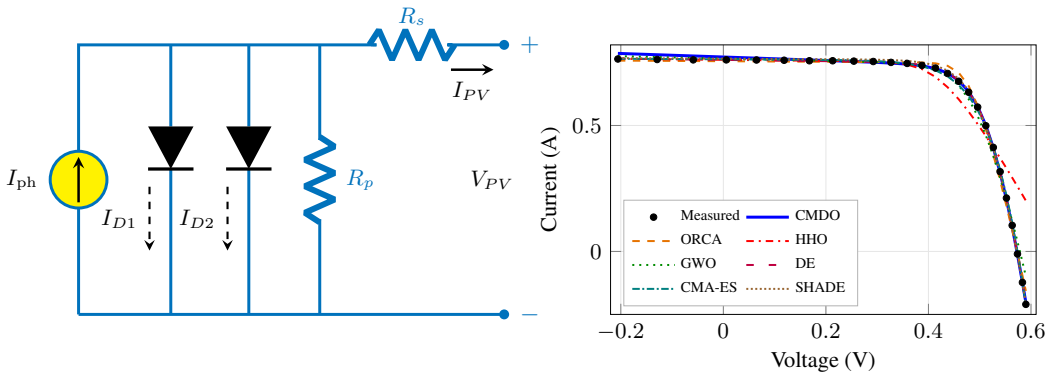
We further evaluate CMDO on the RTC France two-diode photovoltaic parameter-estimation problem using the measured current–voltage data reported by Muhammad et al. (2019). The experiment uses 26 measured I–V observations collected at $33^{\circ}\mathrm{C}$ and $1000\ \mathrm{W/m^2}$. All compared methods use the same measured data, parameter bounds, objective definition, and budget of 2000 objective evaluations per run. Results are reported over seeds 101–110.

Figure 3a illustrates the equivalent circuit used in the experiment. The seven optimized parameters are I_{ph}, I_{01}, I_{02}, R_s, R_p, a_1, and a_2. For each candidate parameter vector, the two-diode model is evaluated at the measured voltage points and the mean absolute error between measured and modeled current is used as the optimization objective.

Figure 3b compares the measured RTC France I–V observations with fitted responses produced by CMDO and the six comparison methods. For each optimizer, the displayed curve corresponds to the run closest to that method's ten-seed median MAE, providing a representative visualization of its fitted parameter set. The curves reproduce the overall nonlinear shape of the measured response to different degrees, with the most visible deviations occurring around the knee and high-voltage region of the characteristic.

Table 2 summarizes the corresponding optimization performance over the ten runs. CMDO obtains a median MAE of 0.005402 A, a mean MAE of 0.010135 A, and a median RMSE of 0.007720 A. Its median MAE is lower than those of HHO, GWO, ORCA, and SHADE, while DE and CMA-ES obtain lower median values.The fitted responses in Figure 3b complement these aggregate errors by showing how the estimated model parameters translate into the resulting current–voltage characteristics.

The paired-seed comparison provides an additional view of run-to-run behavior. CMDO obtains a lower MAE than HHO in all ten paired runs, ORCA in eight, GWO in seven, and both CMA-ES and SHADE in six. Against DE, CMDO obtains the lower MAE in two of the ten runs. CMA-ES has a slightly lower median MAE, while CMDO obtains lower MAE in six paired seeds, illustrating



(a) Two-diode equivalent circuit.

(b) Fitted I–V characteristics.

Figure 3: Photovoltaic parameter-estimation setup and response: (a) equivalent circuit of the seven-parameter two-diode model, and (b) measured versus fitted I–V characteristics for the RTC France solar cell at 33°C and 1000 $\mathrm{W/m^2}$.

Table 2: RTC France two-diode parameter-estimation results over ten seeds. Errors are in amperes.

Method	**Median MAE**	**Mean MAE**	**Median RMSE**	**CMDO wins**
CMDO	0.005402	0.010135	0.007720	–
HHO	0.067527	0.066728	0.106345	10/10
GWO	0.015631	0.017507	0.025084	7/10
ORCA	0.014460	0.019404	0.020608	8/10
DE	0.001841	0.002706	0.002306	2/10
CMA-ES	0.005338	0.005368	0.006985	6/10
SHADE	0.005724	0.005646	0.006698	6/10

why aggregate statistics and paired comparisons can provide complementary views of optimizer behavior.

The photovoltaic experiment uses the same CMDO memory architecture and search mechanism as the numerical benchmark experiments. No photovoltaic-specific operator or memory rule is introduced. The experiment therefore provides a second application setting in which the context–behavior–outcome memory representation is used to guide optimization of a nonlinear physical model.

4.5 Discussion

The results show that CMDO's main contribution lies in how search experience is used to influence future decisions. Recalled recipes develop different operator preferences from freshly generated recipes, indicating that memory actively reshapes the search rather than serving only as passive storage. The variation across benchmark functions further suggests that this adaptation is context dependent rather than driven by one globally preferred behavior. The use of both positive and negative experience is important to this process. Successful behaviors can gain support in related search states, while unsuccessful behaviors remain available as evidence against repeating similar decisions. Combined with working, episodic, and consolidated memory, this gives CMDO a mechanism for balancing recent experience with more persistent search knowledge. The photovoltaic experiment provides a complementary test outside synthetic benchmarks. The same CMDO memory architecture is applied to the two-diode parameter-estimation problem without introducing a domain-specific search rule, showing that the proposed context–behavior–outcome representation can also guide nonlinear physical parameter estimation. The present evaluation is limited to a compact set of low-dimensional benchmarks and one measured photovoltaic curve. Future work should

examine higher-dimensional, noisy, constrained, dynamic, and multi-objective settings, as well as whether useful experience can be transferred across related optimization tasks. Overall, the findings support the central idea of CMDO, optimization memory can be more useful when it captures *what was tried, under what search condition, and what consequence followed*, allowing past experience to become an active part of search control.

5 CONCLUSION

We introduced Cognitive Memory-Driven Optimization (CMDO), an optimization framework that represents search experience through the relationship between context, behavior, and outcome. Rather than remembering only successful solutions or parameter values, CMDO retrieves relevant past experience and uses it to shape new search actions under similar conditions. Across BBOB, CEC2017, and photovoltaic parameter-estimation experiments, CMDO shows competitive problem-dependent performance while the memory analysis demonstrates that recalled experience changes the distribution of executed search behaviors. The results also show that both successful and unsuccessful experience can contribute to later decisions through contextual retrieval, reinforcement, and consolidation. More broadly, CMDO provides a way to separate the search mechanism itself from the memory that governs when different behaviors should be reused. This makes it possible to study optimization not only in terms of which operator performs well, but also in terms of which experiences remain useful across changing search conditions. Such a perspective may help support more adaptive optimizers in which search behavior is shaped by accumulated experience rather than by fixed heuristics alone. These findings support a broader view of optimization memory as an active search-control mechanism rather than a passive record of previous evaluations. Future work will investigate richer context representations, larger and more diverse optimization settings, and the transfer of useful search experience across related problems.

AI USE STATEMENT

Generative AI tools were used only to assist with language polishing, grammatical correction, and improving the clarity of the manuscript text. They were not used to generate the research idea, design the CMDO method, develop the algorithm, conduct the experiments, select the results, or form the scientific conclusions. All AI-assisted text was reviewed and revised by the authors, who take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We have taken several steps to support reproducibility. The main paper describes the CMDO search process, context representation, memory structure, retrieval mechanism, reward definition, and experimental settings. The appendix provides additional implementation details, hyperparameters, memory update rules, benchmark configurations, and supplementary experimental results. The numerical benchmark experiments specify the tested functions, dimensions, evaluation budgets, seeds, and comparison methods, while the photovoltaic experiment documents the measured dataset, model parameters, objective function, and evaluation protocol. Additional implementation details and supporting experimental material are included in the appendix and supplementary materials where permitted.

REFERENCES

N. Awad, M. Ali, J. Liang, B. Qu, and P. Suganthan. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. Technical report, Technical Report, 2016.

Charles Blundell, Benigno Uria, Alexander Pritzel, Yazhe Li, Avraham Ruderman, Joel Z. Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. Model-free episodic control, 2016.

Aydar Bulatov, Yuri Kuratov, and Mikhail S. Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*, 2022.

Rafet Durgut, Mehmet Emin Aydin, and Ibrahim Atli. Adaptive operator selection with reinforcement learning. *Information Sciences*, 581:773–790, 2021. doi: 10.1016/j.ins.2021.10.025.

Noorbakhsh Amiri Golilarz, Hui Gao, Abdoljalil Addeh, and Saeid Pirasteh. ORCA optimization algorithm: A new meta-heuristic tool for complex optimization problems. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2020. doi: 10.1109/ICCWAMTIP51612.2020.9317473.

Noorbakhsh Amiri Golilarz, Hassan S. Al Khatib, and Shahram Rahimi. Toward neurocognitive-inspired intelligence: From AI's structural mimicry to human-like functional cognition. *IEEE Access*, 14:67622–67648, 2026. doi: 10.1109/ACCESS.2026.3689754.

Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11:1–18, 2003.

Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):114–144, 2021. doi: 10.1080/10556788.2020.1808977.

Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, and Huiling Chen. Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97:849–872, 2019. doi: 10.1016/j.future.2019.02.028.

Hao Hu, Jianing Ye, Guangxiang Zhu, Zhizhou Ren, and Chongjie Zhang. Generalizable episodic memory for deep reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.

Gavin Money, Sindhuja Penchala, Jiacheng Li, and Noorbakhsh Amiri Golilarz. Where to bind matters: Hebbian fast weights in vision transformers for few-shot character recognition. In *Proceedings of the 18th IEEE International Conference on Computational Intelligence and Communication Networks (CICN)*, 2026.

F. F. Muhammad, A. W. Karim Sangawi, S. Hashim, S. K. Ghoshal, I. K. Abdullah, and S. S. Hameed. Simple and efficient estimation of photovoltaic cells and modules parameters using approximation and correction technique. *PLOS ONE*, 14(5):e0216201, 2019. doi: 10.1371/journal.pone.0216201.

Mohammed Yusuf Mujawar and Noorbakhsh Amiri Golilarz. Adaptive hebbian memory routing in vision transformers for few-shot learning, 2026a.

Mohammed Yusuf Mujawar and Noorbakhsh Amiri Golilarz. Where should experience live? hierarchical hebbian memory for continual vision transformers, 2026b.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adrià Puigdomènech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2827–2836, 2017.

Cheng Qin, Jianing Li, Chen Yang, Bin Ai, and Yecheng Zhou. Comparative study of parameter extraction from a solar cell or a photovoltaic module by combining metaheuristic algorithms with different simulation current calculation methods. *Energies*, 17(10):2284, 2024. doi: 10.3390/en17102284.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020.

Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. In *International Conference on Learning Representations*, 2019.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

Ryoji Tanabe. Analyzing adaptive parameter landscapes in parameter adaptation methods for differential evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2020. doi: 10.1145/3377930.3389820.

Ryoji Tanabe and Alex Fukunaga. Success-history based parameter adaptation for differential evolution. In *2013 IEEE Congress on Evolutionary Computation*, pp. 71–78, 2013.

Ryoji Tanabe and Alex Fukunaga. Reviewing and benchmarking parameter control methods in differential evolution, 2020.

Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations*, 2022.

A Appendix

A CMDO Algorithmic Details

This appendix provides implementation details for CMDO that complement the higher-level description in Section 3. The algorithm operates in a normalized search space $[0, 1]^D$, while objective evaluations are performed after mapping candidate solutions to the physical problem bounds. All function evaluations, including initialization, count toward the evaluation budget.

A.1 Overall CMDO procedure

Algorithm 1 summarizes the complete optimization loop. At each step, one focal population member is selected cyclically. CMDO describes the current search state, retrieves relevant experience, selects either a recalled or fresh recipe, constructs one candidate, evaluates it, and records the resulting experience. A recalled recipe does not replay an old candidate vector. The recipe specifies a form of search behavior that is reconstructed using the current population. Consequently, the same remembered recipe can produce different displacements when it is reused in different search states.

A.2 Search context

For focal solution $\mathbf{z}_i$, CMDO uses the eight-dimensional context

$$\mathbf{c}_i = [c_i^{(1)}, \ldots, c_i^{(8)}] \tag{13}$$

The components represent population spread, focal rank, distance to the current best, distance to the population centroid, consumed evaluation budget, stagnation, recent positive-reward frequency, and

focal-to-best fitness gap. The implementation computes them as

$$c_i^{(1)} = \min\left(1,\ 2\ \mathrm{mean}_d[\mathrm{std}(Z_{:,d})]\right), \tag{14}$$

$$c_i^{(2)} = \frac{\sum_{j=1}^{N} \mathbb{I}[f_j < f_i]}{\max(1, N-1)}, \tag{15}$$

$$c_i^{(3)} = \frac{\|\mathbf{z}_i - \mathbf{z}_{best}\|_2}{\sqrt{D}}, \tag{16}$$

$$c_i^{(4)} = \frac{\|\mathbf{z}_i - \bar{\mathbf{z}}\|_2}{\sqrt{D}}, \tag{17}$$

$$c_i^{(5)} = \frac{n_{\mathrm{eval}}}{B}, \tag{18}$$

$$c_i^{(6)} = \min\left(1,\ \frac{s_{\mathrm{stag}}}{10N}\right), \tag{19}$$

$$c_i^{(7)} = \frac{1}{|H|} \sum_{R \in H} \mathbb{I}[R > 0], \tag{20}$$

$$c_i^{(8)} = \tanh\left(\frac{\max(0, f_i - f_{best})}{\max(\mathrm{MAD}(\mathbf{f}), 10^{-12})}\right) \tag{21}$$

Here H denotes the recent reward history. The true optimum of a benchmark function is not used in the context or search controller.

A.3 Fresh recipes and search geometry

A recipe is

$$\mathbf{r} = (o, s, m, g), \tag{22}$$

where o selects exploration, directed search, or local refinement; s is the step scale; m controls the directional mixture; and g specifies full-space or subspace geometry.

Fresh recipes use

$$P(o = \text{exploration}) = 0.5, \qquad P(o = \text{directed}) = 0.3, \qquad P(o = \text{local}) = 0.2, \tag{23}$$

with

$$s \sim \mathrm{LogUniform}(0.25, 2), \qquad m \sim \mathcal{U}(0.2, 0.8) \tag{24}$$

For two sampled population members,

$$\boldsymbol{\delta} = \mathbf{z}_a - \mathbf{z}_b, \tag{25}$$

and the stochastic component is

$$\boldsymbol{\epsilon} = \boldsymbol{\eta} \odot \max(\boldsymbol{\sigma}_Z, 0.01), \qquad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{26}$$

The three directions are

$$\mathbf{d}_{exp} = m\boldsymbol{\delta} + (1-m)\boldsymbol{\epsilon}, \tag{27}$$

$$\mathbf{d}_{dir} = m(\mathbf{z}_{best} - \mathbf{z}_i) + (1-m)\boldsymbol{\delta}, \tag{28}$$

$$\mathbf{d}_{loc} = 0.1\left[m(\mathbf{z}_{best} - \mathbf{z}_i) + (1-m)\boldsymbol{\epsilon}\right] \tag{29}$$

For subspace geometry and $D > 1$, CMDO modifies

$$\left\lceil \sqrt{D} \right\rceil \tag{30}$$

coordinates sampled uniformly without replacement. Coordinates outside the selected subset remain equal to those of the focal solution.

Algorithm 1 Cognitive Memory-Driven Optimization (CMDO)

Require: Objective f, bounds $(\mathbf{l}, \mathbf{u})$, dimension D, evaluation budget B, population size N
Ensure: Best solution $\mathbf{x}_{best}$

Initialize N solutions uniformly in $[0,1]^D$
Evaluate the initial population and count all evaluations
Initialize working memory $\mathcal{M}_W$, episodic memory $\mathcal{M}_E$, and consolidated memory $\mathcal{M}_C$
Initialize reward history and stagnation counter
while $n_{\text{eval}} < B$ **do**
    Select focal index i cyclically
    Compute search context $\mathbf{c}_i$
    Retrieve contextually relevant records from $\mathcal{M}_W \cup \mathcal{M}_E \cup \mathcal{M}_C$
    if no usable recalled recipe exists **or** fresh exploration is selected **then**
        Generate a fresh recipe $\mathbf{r} = (o, s, m, g)$
        Set recalled source to
    else
        Select a recalled recipe according to memory support
        Record its source experience
    end if
    Construct search direction $\mathbf{d}$ from the current population
    Apply full-space or subspace geometry according to g
    Generate candidate $\mathbf{z}' = \text{Reflect}(\mathbf{z}_i + s\mathbf{d})$
    Map $\mathbf{z}'$ from $[0,1]^D$ to the physical search space
    Compute pre-evaluation fitness scale $\sigma_f = \max(\text{MAD}(\mathbf{f}), 10^{-12})$
    Evaluate $f(\mathbf{x}')$
    Compute $R = \tanh((f_i - f')/\sigma_f)$
    if $f' < f_i$ **then**
        Replace the focal solution with the candidate
    end if
    Update the global best and stagnation state
    if a recalled source was executed **then**
        Update reuse count and reuse return of that source only
    end if
    Store the new experience in working and episodic memory
    Consolidate eligible repeatedly useful experience
    Apply bounded-memory retention and eviction
end while
return $\mathbf{x}_{best}$

A.4 Retrieval and negative evidence

Records from the three memory stores are first deduplicated by experience identifier. Similarity between the current context $\mathbf{c}$ and stored context $\mathbf{c}_j$ is

$$S_j = \exp\left(-\frac{\text{MSE}(\mathbf{c}, \mathbf{c}_j)}{2h^2}\right), \tag{31}$$

and the age-adjusted retrieval weight is

$$W_j = S_j \rho^{a_j} \tag{32}$$

Records below the relevance threshold are removed, after which the top-k records by retrieval weight are considered. The current value of a record is

$$V_j = \frac{R_j + R_j^{\text{reuse}}}{1 + n_j^{\text{reuse}}} \tag{33}$$

Only positive-valued records can directly propose a recalled recipe. Retrieved negative records using the same operator and geometry contribute a penalty

$$P_j = \sum_{q \in \mathcal{N}_j} W_q |V_q| \exp\left(-\left|\log\frac{s_q}{s_j}\right| - |m_q - m_j|\right), \tag{34}$$

Table 3: Numerical benchmark configuration.

Suite	**Functions**	D	**Budget**	**Seeds**
BBOB/COCO	F1, F8, F15	5	500	15–17
CEC2017	F1, F4, F10	10	1000	15–17

giving the final support

$$Q_j = W_j V_j - \lambda P_j. \tag{35}$$

Recipes with $Q_j > 0$ are sampled proportionally to their support. If none remain, a fresh recipe is used.

A.5 Memory update and consolidation

Each completed search step generates a new experience containing the search context, executed recipe, reward, search step, and realized displacement, together with metadata used for later reuse. The new event is written to both working and episodic memory, regardless of whether its reward is positive or negative.

When a remembered recipe is actually executed, reuse feedback is assigned only to the selected source record. An original experience becomes eligible for consolidation after at least three selected reuses with a positive mean reuse return.

Two records are compatible for consolidation when they have the same operator and geometry and satisfy

$$\mathrm{MSE}(\mathbf{c}_p, \mathbf{c}_j) < 0.01, \tag{36}$$

$$\left|\log \frac{s_p}{s_j}\right| < 0.25, \qquad |m_p - m_j| < 0.1 \tag{37}$$

Compatible records update a consolidated prototype; otherwise, a new prototype is formed. Working memory retains the newest 24 experiences, episodic memory is bounded at 256 records using age-decayed utility for eviction, and consolidated memory contains at most 24 prototypes.

B Experimental Implementation Details

B.1 Benchmark protocol

The numerical experiments use BBOB/COCO functions F1, F8, and F15 at $D = 5$ with 500 evaluations, and CEC2017 functions F1, F4, and F10 at $D = 10$ with 1000 evaluations. Instance 1 and seeds 15–17 are used for the reported benchmark panel. All objective evaluations, including initialization, count toward the stated budget.

C Additional Convergence Results

The main paper presents CEC2017 F10 as a representative convergence example. Figure 4 shows the remaining recorded convergence trajectories together with F10 for completeness. Curves report the median best-so-far error across seeds 15–17.

These trajectories complement the final-error tables by showing when improvements occur within the fixed evaluation budget. They also illustrate the problem-dependent evolution of CMDO's search behavior across the selected landscapes.

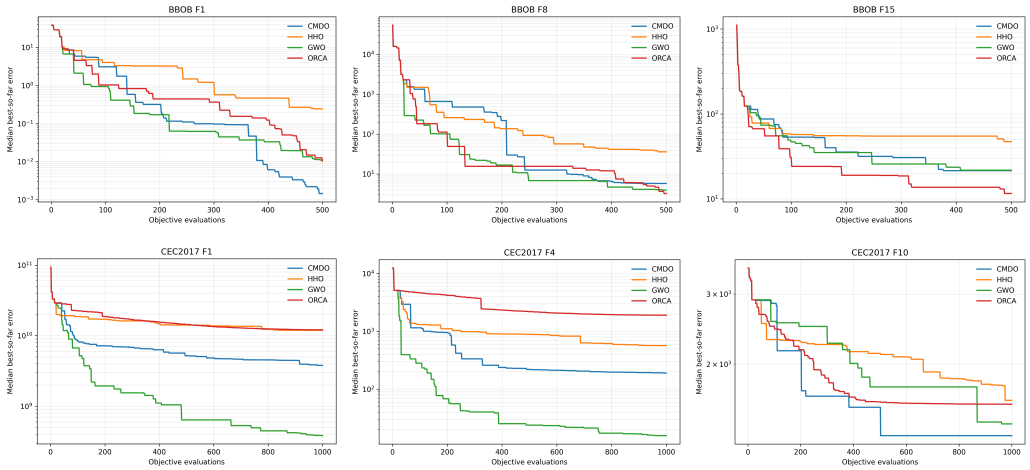


Figure 4: Median best-so-far optimization error over seeds 15–17. The top row shows BBOB F1, F8, and F15; the bottom row shows CEC2017 F1, F4, and F10. The vertical axis is logarithmic.

D Photovoltaic Model and Experimental Details

D.1 Two-diode photovoltaic model

The photovoltaic experiment estimates the seven parameters

$$\boldsymbol{\theta} = [I_{\mathrm{ph}}, I_{01}, I_{02}, R_s, R_p, a_1, a_2] \tag{38}$$

of the two-diode model. For measured terminal voltage V and current I, the model is defined implicitly as

$$I = I_{\mathrm{ph}} - I_{01}\left[\exp\left(\frac{V + IR_s}{a_1 V_T}\right) - 1\right] - I_{02}\left[\exp\left(\frac{V + IR_s}{a_2 V_T}\right) - 1\right] - \frac{V + IR_s}{R_p}, \tag{39}$$

where

$$V_T = \frac{N_s k T}{q} \tag{40}$$

Here k is the Boltzmann constant, q is the elementary charge, T is the cell temperature in Kelvin, and $N_s = 1$ for the RTC France cell.

The optimization objective is

$$\mathrm{MAE}(\boldsymbol{\theta}) = \frac{1}{26}\sum_{j=1}^{26}\left|I_j^{\mathrm{meas}} - I_j^{\mathrm{model}}(\boldsymbol{\theta})\right| \tag{41}$$

RMSE is calculated after optimization as an additional fit-quality measure.

D.2 Parameter bounds and representation

The search is performed in a common normalized seven-dimensional space. I_{01} and I_{02} are decoded logarithmically because their ranges span multiple orders of magnitude; the remaining parameters are mapped linearly.

The experiment uses the 26 measured RTC France I–V observations at $33^{\circ}\mathrm{C}$ and $1000\ \mathrm{W/m^2}$ reported by Muhammad et al. (2019). Parameter bounds follow the RTC France two-diode settings used by Qin et al. (2024). Each optimizer receives 2000 objective evaluations for seeds 101–110.

For every candidate parameter vector, the implicit model equation is solved at all measured voltages before MAE is computed. The same model solver, measured data, parameter encoding, and objective function are used for all compared optimization methods.

Table 4: Parameter bounds for the RTC France two-diode model.

Parameter	**Lower bound**	**Upper bound**
I_{ph} (A)	0	1
I_{01} (A)	10^{-15}	10^{-3}
I_{02} (A)	10^{-15}	10^{-3}
R_s (Ω)	0	0.5
R_p (Ω)	0.001	100
a_1	0.5	5
a_2	1	5