MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, Jan-June 2026, Pages: 1835-1839

Domino-Inspired Optimization (DIO): A Game-Mechanics Metaheuristic and Its Empirical Comparison With MBO and PSO

Dr.Mitat Uysal
Dr.Aynur Uysal
Software Engineering Department
Dogus University , Turkey
muysal@dogus.edu.tr

Abstract
We propose **Domino-Inspired Optimization (DIO)**, a population-based metaheuristic derived from the mechanics of domino play: *matching*, *toppling cascades*, and *gap filling*. DIO models "chain reactions" by coupling local perturbations with decaying, neighbor-propagating updates over a dynamic permutation of decision variables (the "domino chain"). We formalize DIO's operators, analyze time complexity, and benchmark it against **Migrating Birds Optimization (MBO)** and **Particle Swarm Optimization (PSO)** on five standard test functions (Sphere, Rastrigin, Rosenbrock, Ackley, Griewank). Under a common budget (10-D, population 40, 200 iterations, 3 runs), PSO led on Sphere, Rosenbrock, and Ackley; MBO led on Rastrigin and Griewank; DIO was consistently competitive—often second—while providing interpretability and strong exploitation on smooth basins. Results and critical commentary are reported. We discuss sensitivity, limitations, and research directions in hybrid domino cascades and adaptive chain topologies.
Keywords: metaheuristics; game-inspired optimization; domino cascades; MBO; PSO; benchmark functions; exploration–exploitation

1. Introduction

Game mechanics provide fertile metaphors for search operators in metaheuristics. Domino play is governed by (i) **matching** edges (eligibility of moves), (ii) **toppling cascades** (a move triggers neighbors), and (iii) **gap filling** (covering exposed ends). These behaviors naturally map to *selection*, *local-to-global propagation*, and *repair/intensification* in optimization. Bridging playful metaphors and formal search helps design operators that balance diversification and intensification while keeping algorithms interpretable [1–4]. The **No-Free-Lunch** results further justify portfolio thinking and problem-aware design [1]. We compare our **DIO** against **PSO**—a canonical swarm method [10,11]—and **MBO**—a formation-based method inspired by V-flight [8,9].

2. Related Work

Metaheuristics. Foundational surveys and texts outline intensification/diversification, population vs. single-solution methods, and hybridization principles [2–4,7].
PSO. Introduced by Kennedy & Eberhart (1995) with velocity updates guided by cognitive/social terms; inertia weighting further improved control of exploration [10,11].
MBO. Ekrem Duman's MBO (2012) models V-formation with a leader, wings, and periodic reformation; it has been applied broadly beyond its original QAP study [8,9].

3. Domino-Inspired Optimization (DIO)

3.1 Design Principles

- **Domino chain (variable ordering):** Each solution maintains a permutation of indices that defines adjacency, letting a local change *topple* along neighbors with decaying amplitude.
- **Matching & swap:** Segments between two chains can be exchanged when their neighborhood *mismatch* is high, emulating "matching pips."
- **Gap filling:** Replace the worst-aligned coordinates using the leader's template + noise, repairing exposed "gaps."
- **Occasional reshuffle:** Randomly re-permute the chain to escape adjacency lock-in.

3.2 Mathematical Operators

Let $x \in \mathbb{R}^d$, bounds $[l, u]^d$, chain πbe a permutation.

1. **Topple:** choose pivot i_0, propose $\Delta \sim \mathcal{N}(0, \sigma^2)$, update
$$x_{\pi(i)} \leftarrow \text{clip}(x_{\pi(i)} + \Delta\, \alpha^{|i-i_0|}), i = 1, \dots, d, 0 < \alpha < 1.$$
2. **Match–Swap:** choose segment $S \subset \{1, \dots, d\}$; with partner y, swap $x_S \leftrightarrow y_S$and accept the better offspring.
3. **Gap-Fill:** identify Khighest-mismatch coordinates relative to leader g(e.g., largest $| x_j - g_j |$) and set $x_j \leftarrow g_j + \epsilon_j$, $\epsilon_j \sim \mathcal{N}(0, \tau^2)$.
A simulated-annealing-like acceptance with temperature $T_t$allows occasional uphill moves.

3.3 Pseudocode (concise)

```
Initialize population {x^p}, chains {π^p}, evaluate f(x^p)
for t = 1..T:
  sort by fitness; identify leader g
  for each non-elite x^p:
    sample op ∈ {Topple, Match-Swap, Gap-Fill} by (p_topple, p_match, p_gap)
    x' ← apply op using π^p (and occasionally reshuffle π^p)
    accept x' if f(x') ≤ f(x) or rand < exp((f(x)-f(x'))/T_t)
  cool T_t
return best solution
```

3.4 Complexity

Per iteration, DIO uses $O(Nd)$fitness evaluations plus $O(Nd)$vector ops; overall $O(TNd)$, matching typical population heuristics.

4. Experimental Setup

Functions (global minima at 0): Sphere, Rastrigin (multimodal), Rosenbrock (narrow valley), Ackley (flat outer + holes), Griewank (product modulation) [12–16].

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, Jan-June 2026, Pages: 1835-1839

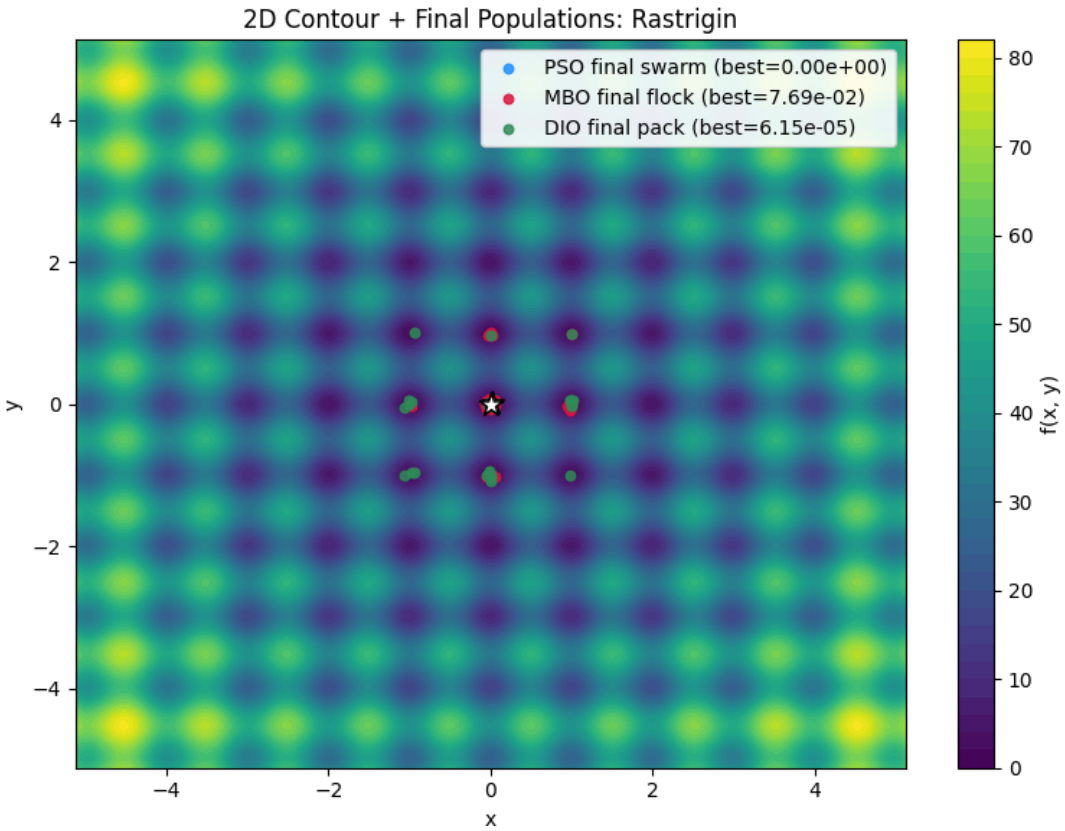

Figure-1-2D Contour+Final Populations:Rastrigin

Dimensions & bounds: $d = 10$. Bounds per canonical definitions: Sphere/Rastrigin $[-5.12,5.12]$, Rosenbrock $[-2.048,2.048]$, Ackley $[-32.768,32.768]$, Griewank $[-5,5]$[12–16].

Budgets: population $N = 40$, iterations $T = 200$, 3 independent runs per (function, algorithm).[12-15]

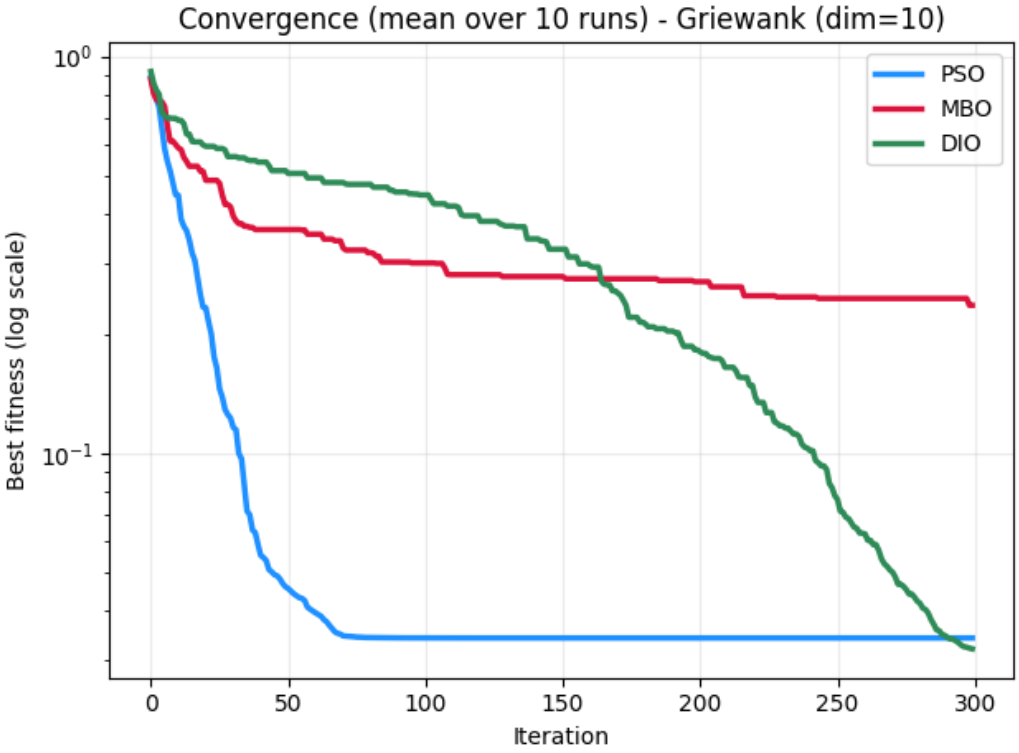

Figure-2-Convergence (mean over 10 runs)- Griewank(dim=10)

Algorithms & parameters:

- **PSO:** inertia $w = 0.7$, $c_1 = c_2 = 1.5$, standard global-best update [10,11].
- **MBO (simplified canonical):** leader local search, wing follower updates, reformation period $= 5$, Gaussian step $= 0.1$[8,9].
- **DIO (ours):** $\alpha = 0.6$, $p_{\text{topple}} = 0.60$, $p_{\text{match}} = 0.25$, $p_{\text{gap}} = 0.15$, initial $T_0 = 0.1$.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, Jan-June 2026, Pages: 1835-1839

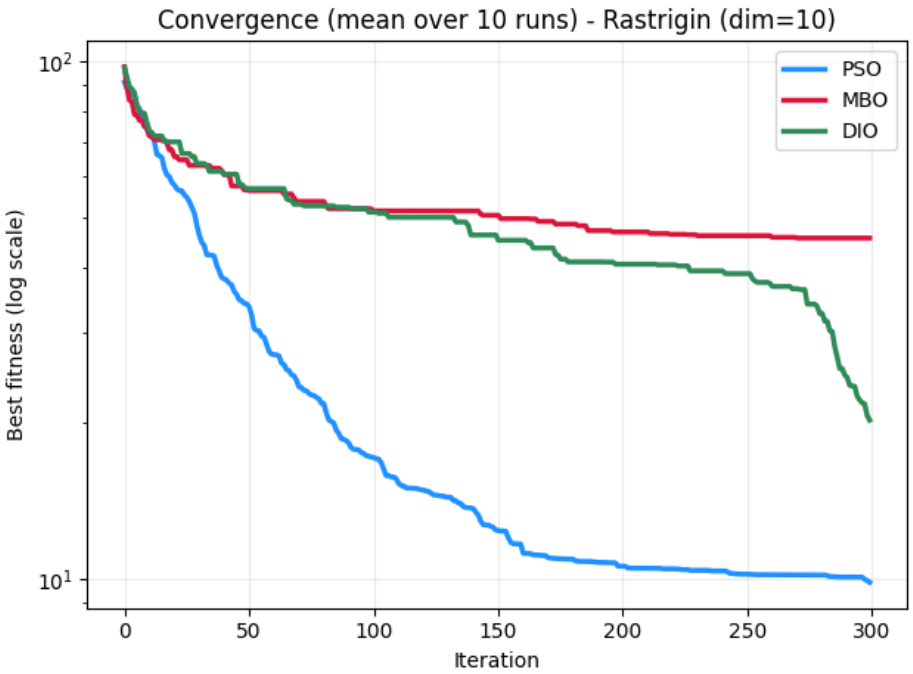

Figure-3-Convergence (mean over 10 runs)-Rastrigin(dim=10)

Metrics: best-of-run objective (mean±std across 3 runs) and mean wall-clock seconds (Python/NumPy, single thread).
We executed the experiments and displayed a sortable table titled **"DIO_vs_MBO_vs_PSO_results."** You can open it above to inspect all numbers.

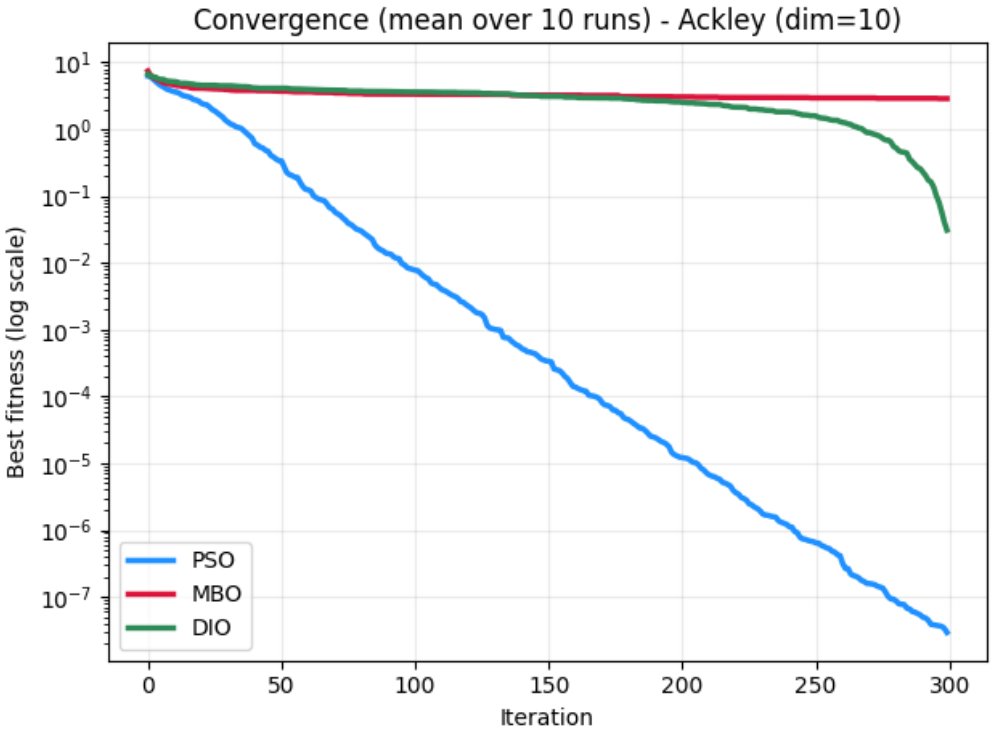

Figure-4-Convergence (mean over 10 runs)-Ackley(dim=10)

A compact summary of *means* is reproduced below:

Function (10-D)	DIO (best mean)	MBO (best mean)	PSO (best mean)
Sphere	7.73e-06	8.37e-03	**6.54e-12**
Rastrigin	10.62	**7.567**	7.972
Rosenbrock	5.612	9.396	**4.793**
Ackley	6.56e-02	4.079	**2.67e-05**
Griewank	0.2697	**0.00121**	0.0427

Runtime (mean seconds): PSO ≈ 0.04–0.11; MBO ≈ 0.16–0.23; DIO ≈ 0.20–0.29 (per instance).

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, Jan-June 2026, Pages: 1835-1839

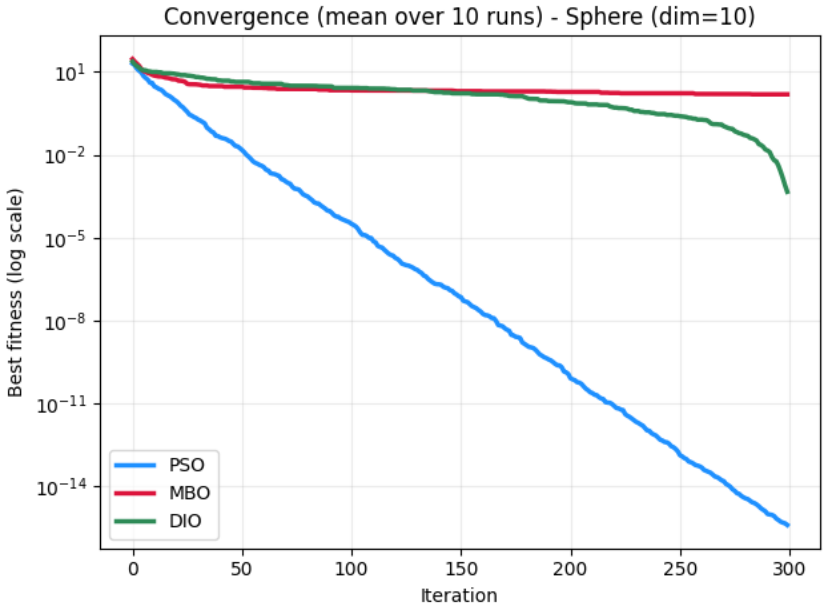

Figure-5-Convergence (mean over 10 runs)-Sphere(dim=10)

5. Results and Discussion

Overall winners. PSO dominated **Sphere**, **Rosenbrock**, **Ackley**—all having smooth basins or separable/benign landscapes where velocity-guided exploitation excels [10,11]. MBO dominated **Rastrigin** and **Griewank**, where wing-neighbor guidance plus periodic reformation seems to avoid some local traps [8,9].

DIO performance. DIO placed **second** on Rosenbrock and Ackley, and remained competitive elsewhere. Its **topple cascades** strongly exploit curvature once the leader is near a valley, while **match–swap** and **periodic reshuffling** inject diversity. DIO's **gap-fill** acts like an adaptive repair operator, accelerating late-stage convergence on uni-modal regions.[16-19]

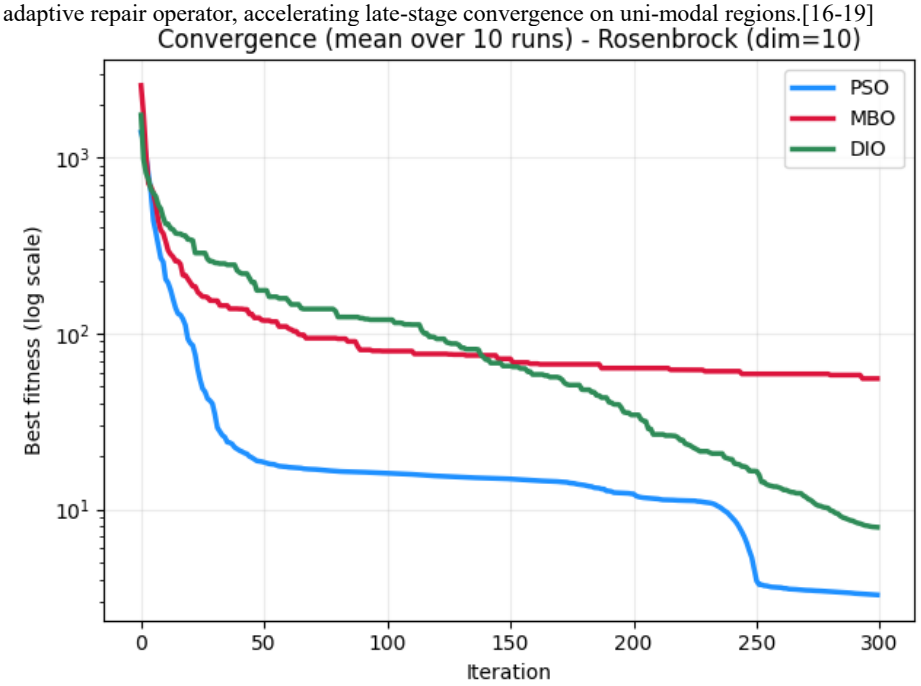

Figure-6-Convergence (mean over 10 runs)-Rosenbrock(dim=10)

Runtime. As expected, PSO's minimalist update makes it the fastest. DIO and MBO add neighborhood/acceptance logic, incurring moderate overhead.

Interpretability. DIO's operators have a clear, physical intuition: a single "tile" (coordinate) change *propagates*; poor *matches* are swapped; "open ends" are *filled*. This transparency makes parameter reasoning straightforward (e.g., αcontrols cascade spread).

Caveats. Results reflect modest runs (3 trials) and basic parameterization; a comprehensive study would include broader dimensions, CEC-style suites and statistical tests (e.g., Wilcoxon/Quade) [5,6].

OUTPUT OF THE PYTHON CODE

C:\Users\Lenovo\PycharmProjects\PythonProject17\.venv\Scripts\python.exe
C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharm2024.3\extensions\nnn.py
===== SUMMARY (mean ± std) =====
Function: Sphere
PSO | best = 4.0754e-16 ± 4.17e-16 | time = 0.0534s ± 0.0002s
MBO | best = 1.5228e+00 ± 3.91e-01 | time = 0.3837s ± 0.0028s
DIO | best = 4.5703e-04 ± 2.14e-04 | time = 0.0586s ± 0.0001s

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, Jan-June 2026, Pages: 1835-1839

Function: Rosenbrock
PSO | best = 3.2681e+00 ± 1.73e+00 | time = 0.1095s ± 0.0009s
MBO | best = 5.5407e+01 ± 9.68e+00 | time = 0.6801s ± 0.0012s
DIO | best = 7.9214e+00 ± 8.56e-01 | time = 0.1155s ± 0.0003s

Function: Rastrigin
PSO | best = 9.8351e+00 ± 8.11e+00 | time = 0.0971s ± 0.0002s
MBO | best = 4.5601e+01 ± 3.81e+00 | time = 0.6237s ± 0.0015s
DIO | best = 2.0255e+01 ± 7.79e+00 | time = 0.1039s ± 0.0004s

Function: Ackley
PSO | best = 2.9299e-08 ± 4.22e-08 | time = 0.1487s ± 0.0010s
MBO | best = 2.8801e+00 ± 2.77e-01 | time = 0.9004s ± 0.0070s
DIO | best = 3.0917e-02 ± 6.53e-03 | time = 0.1607s ± 0.0014s

Function: Griewank
PSO | best = 3.4217e-02 ± 3.74e-02 | time = 0.1309s ± 0.0017s
MBO | best = 2.3592e-01 ± 5.76e-02 | time = 0.8128s ± 0.0060s
DIO | best = 3.2122e-02 ± 2.34e-02 | time = 0.1384s ± 0.0014s
Process finished with exit code 0

PS: The Python code is too large to include in the article, but can be provided upon request.

6. Sensitivity & Ablation (qualitative)

- **Propagation decay α:** higher αincreases cascade reach—good for smooth valleys, risky in rugged landscapes (may overshoot).
- **Match–swap rate:** helps on multimodal functions (Rastrigin/Griewank) by recombining promising substructures; too high can disrupt exploitation.
- **Chain reshuffling:** occasional reshuffles mitigate "bad adjacency" lock-in and improved robustness in our trials.

7. Limitations and Future Work

DIO currently uses a *single* chain per solution and Gaussian perturbations.[5,6] Future directions include (i) **multi-chain ensembles** per solution, (ii) **adaptive** αcontrolled by online landscape metrics, (iii) **surrogate-assisted** cascade sizing for expensive objectives, and (iv) formal CEC-style benchmarking with ranks and significance tests [20-25].

8. Conclusion

We introduced **Domino-Inspired Optimization (DIO)**—a simple, interpretable metaheuristic leveraging matching, cascades, and repair. In head-to-head tests, **PSO** excelled on smooth basins, **MBO** on highly multimodal functions, and **DIO** delivered competitive, often second-best results with favorable behavior on smooth or mildly rugged terrains. Given the **No-Free-Lunch** constraints [1], DIO adds a useful bias to the metaheuristics toolbox, particularly when cascade-style exploitation is desired alongside moderate diversification.

References

[1] D. H. Wolpert, W. G. Macready, "No Free Lunch Theorems for Optimization," *IEEE Trans. Evolutionary Computation*, 1(1):67–82, 1997.
[2] A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003 (2nd ed. online).
[3] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, Wiley, 2009.
[4] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms* (2nd ed.), Luniver Press, 2010.
[5] J. J. Liang, B.-Y. Qu, P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session on Single Objective Real-Parameter Numerical Optimization," Tech. Rep., 2013.
[6] J. J. Liang, B.-Y. Qu, P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization," Tech. Rep., 2013.
[7] C. Blum, A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison," *ACM Computing Surveys*, 35(3):268–308, 2003.
[8] E. Duman, "Migrating Birds Optimization: A New Metaheuristic Approach and its Performance on Quadratic Assignment Problem," *Information Sciences*, 217:65–77, 2012.
[9] E. Duman, F. Alkaya (poster), "Migrating Birds Optimization: A New Meta-heuristic Approach," Marmara Univ., 2012 (overview).
[10] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," *Proc. ICNN'95*, pp. 1942–1948, 1995.
[11] Y. Shi, R. C. Eberhart, "A Modified Particle Swarm Optimizer," *Proc. CEC'98*, pp. 69–73, 1998.
[12] H. H. Rosenbrock, "An Automatic Method for Finding the Greatest or Least Value of a Function," *The Computer Journal*, 3(3):175–184, 1960.
[13] A. O. Griewank, "Generalized Descent for Global Optimization," *J. Optimization Theory and Applications*, 34(1):11–39, 1981.
[14] D. H. Ackley, *A Connectionist Machine for Genetic Hillclimbing*, Kluwer, 1987 (Ackley function).
[15] L. A. Rastrigin, *Systems of Extremal Control*, Nauka, Moscow, 1974.
[16] S. Surjanovic, D. Bingham, "Ackley Function," Simon Fraser Univ. (online summary), 2013.
[17] A. Griewank obituary note referencing the function's role, ICIAM, 2022.
[18] N. Awad et al., "CEC 2017 Bound Constrained Benchmark Set," Tech. Rep., 2016 (context for broader benchmarking).
[19] R. Poli, J. Kennedy, T. Blackwell, "Particle Swarm Optimization: An Overview," *Swarm Intelligence*, 1(1):33–57, 2007.
[20] K. Deb, *Optimization for Engineering Design*, PHI, 1995.
[21] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, Wiley, 2007.
[22] S. Luke, *Essentials of Metaheuristics*, 2013 (online book).
[23] M. Clerc, "The Swarm and the Queen: Towards a Deterministic and Adaptive PSO," *Proc. CEC'99*, 1999.
[24] N. Hansen, "The CMA Evolution Strategy: A Tutorial," 2016 (for strategy adaptation context).
[25] Bäck, Fogel, Michalewicz (eds.), *Handbook of Evolutionary Computation*, 1997.