MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6220-6226

A Churchill-Inspired Metaheuristic Optimization Algorithm: CIRSO (Churchillian Resilience & Strategy Optimizer)

Mitat Uysal, S. Aynur Uysal
Department of Software Engineering, Dogus University,
Istanbul,Turkey

Abstract:
This paper proposes a new metaheuristic optimizer inspired by Winston Churchill's strategic leadership attributes: resilience under adversity, coalition-building, disciplined yet adaptive decision-making, and the ability to "rally" effort when progress stalls. We translate these attributes into algorithmic operators that combine (i) persistence against stagnation, (ii) cabinet-style multi-policy exploration, (iii) coalition fusion around elite solutions, and (iv) morale-boost "rally steps" that intensify local exploitation near promising regions. The proposed method—CIRSO—is evaluated on a convex quadratic test problem $z = (x-3)^2 + (y-2)^2$ and five standard benchmark functions. A complete Python implementation (NumPy + Matplotlib only) is provided with multiple graphical outputs (colored convergence curves, 2D contours with trajectories, 3D surface paths, and multi-run statistics). The results show that Churchill-inspired operators can improve robustness against premature convergence while maintaining competitive convergence speed.
Keywords: Metaheuristic optimization; leadership-inspired algorithms; coalition search; benchmark functions; Churchill.

I. Introduction

Metaheuristic optimization methods search for near-global solutions in complex, non-convex, and non-differentiable spaces without requiring gradient information. Rather than guaranteeing strict optimality, they emphasize robustness, flexibility, and practical performance across diverse problem landscapes. Classical examples include Genetic Algorithms (GA) [1], Simulated Annealing (SA) [2], Particle Swarm Optimization (PSO) [3], Differential Evolution (DE) [4], and Artificial Bee Colony (ABC) [5]. The No-Free-Lunch theorem formally states that no single optimizer outperforms all others across every possible problem, thereby motivating the continuous development of new hybrid, adaptive, and problem-agnostic strategies [6].
In recent years, a growing research direction has focused on concept-inspired optimization, where structured human, biological, artistic, or organizational behaviors are abstracted into computational search operators. Examples include art-inspired frameworks such as Miró-Inspired Metaheuristic Optimization Algorithm, Leonardo-inspired geometric exploration models, Shakespearean Soliloquy Optimization (SSO), and socially coordinated swarm strategies such as Migrating Birds Optimization (MBO) [7],[8],[9],[10]. These works demonstrate that abstract conceptual frameworks can be systematically transformed into effective population-based search dynamics.
Within this broader paradigm, leadership theory offers a largely unexplored yet structurally rich source of inspiration. Winston Churchill (1874–1965), Prime Minister of the United Kingdom during World War II, became a historical symbol of resilience under extreme uncertainty and strategic coalition governance. His leadership emphasized persistence in the face of adversity ("never give in"), multi-perspective deliberation through cabinet debate, adaptive strategic recalibration, and coordinated collective mobilization [11],[12]. These characteristics, when abstracted from their historical context, suggest computational analogues for managing stagnation, balancing exploration and exploitation, and strengthening cooperative elite guidance within population-based optimization.
Churchill's wartime decision-making model can thus be interpreted not merely as political history but as a structured system of adaptive strategy formation under uncertainty. In particular, four principles stand out for algorithmic translation: (i) resistance to stagnation through persistence mechanisms, (ii) multi-policy evaluation analogous to cabinet debate, (iii) coalition-based elite fusion for consensus building, and (iv) morale-driven intensification phases that reinforce promising directions. Unlike purely stochastic restart schemes, such structured resilience mechanisms may provide controlled diversification without discarding accumulated information.Motivated by this perspective, this paper proposes the Churchillian Resilience & Strategy Optimizer (CIRSO), a novel metaheuristic framework that operationalizes leadership-inspired principles into computational operators. CIRSO integrates multi-proposal cabinet moves, elite coalition guidance, structured stagnation detection, and intensification bursts into a unified population-based algorithm. The method aims to address two common limitations of many metaheuristics: premature convergence in local minima and overly rigid single-update rules.
The contributions of this work are threefold:

1. A formal mapping between Churchillian leadership principles and population-based optimization operators.
2. A complete mathematical formulation and pseudocode of the CIRSO algorithm.
3. Experimental validation on a convex quadratic function and five widely used benchmark functions, supported by graphical trajectory analysis and multi-run statistical summaries.

By extending the concept-inspired optimization paradigm into the domain of strategic leadership modeling, this study illustrates how structured human decision frameworks can inform robust algorithmic design.

II. From Churchillian Leadership to Algorithmic Principles

We convert four Churchillian attributes into computational operators:
2.1 Never-Give-In Persistence (Stagnation Resistance) In many population-based optimizers, prolonged periods without improvement indicate convergence to local minima or loss of population diversity. To address this issue, CIRSO explicitly monitors the *stall length*, defined as the number of consecutive iterations without global best improvement. When this value exceeds a predefined threshold, structured *Resilience Actions* are activated to restore exploration capability while preserving learned information.
These resilience mechanisms include:
• *Partial restart of the worst-performing agents*, rather than resetting the entire population
• *Temporary expansion of the exploration radius to enlarge* the search neighborhood
• *Diversification through renewed cabinet proposals*, increasing policy variety within each agent's update step
2.2 War Cabinet Debate (Multi-Policy Proposals) Rather than relying on a single deterministic update rule, CIRSO adopts a *multi-policy evaluation mechanism* inspired by cabinet-style deliberation. For each agent at every iteration, multiple candidate moves are generated in parallel—forming a "cabinet" of strategic proposals. The agent then selects the candidate that yields the greatest objective improvement, ensuring adaptive and performance-driven decision making.
The cabinet proposals consist of:
• *Exploration move:* a stochastic long-range perturbation with controlled scaling to enhance global search capability
• *Exploitation move:* a directed local step toward a selected elite solution to reinforce intensification
• *Compromise move:* a weighted combination of exploration and exploitation components to balance diversification and convergence
2.3 Coalition Government (Elite Fusion) Churchill's coalition governance strategy is translated into CIRSO through an *elite pooling and fusion mechanism*. Instead of relying on a single global best solution, the algorithm maintains a structured elite set that preserves multiple high-quality candidates, thereby reducing premature convergence and maintaining directional diversity.
The coalition mechanism operates as follows:
• *Maintain an elite set E*consisting of the top-performing solutions in the current population
• *Guide new candidate moves using a randomly selected elite member*, which diversifies exploitation pressure across multiple promising regions
• *Perform occasional "coalition merge" operations*, generating consensus solutions by averaging elite members with controlled stochastic perturbation (elite mean + noise)
2.4 Rally the Nation (Intensification Burst) When CIRSO detects significant or sustained improvement in the objective value, it activates a short *rally phase*, designed to intensify the search around promising regions. This mechanism strengthens local refinement without permanently reducing global exploration capacity.
During the rally phase, the algorithm applies the following adjustments:
• *Reduced step size*, enabling finer-grained local search
• *Increased proportion of exploitation-oriented proposals*, prioritizing elite-guided updates
• *Higher probability of local refinement moves*, reinforcing convergence toward the current promising basin

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6220-6226

III. Mathematical Model of CIRSO

Let the objective function be $f(x)$, where $x \in \mathbb{R}^d$. CIRSO maintains a population $\{x_i\}_{i=1}^{N}$ of candidate solutions. At each iteration t, every agent generates a cabinet of candidate proposals, reflecting the multi-policy strategy described in Section 2.

Cabinet proposal set for agent i at iteration t:The exploration proposal introduces stochastic long-range movement:

$$x_i^{(A)} = x_i + s_t \cdot \mathcal{N}(0,1)$$

The elite-guided exploitation proposal directs the search toward a randomly selected elite member $e \in E$, while preserving controlled randomness:

$$x_i^{(B)} = x_i + \alpha_t(e - x_i) + \beta_t \cdot \mathcal{N}(0,1)$$

The compromise proposal balances diversification and intensification by combining the previous two moves:

$$x_i^{(C)} = 0.5\, x_i^{(A)} + 0.5\, x_i^{(B)}$$

Selection Mechanism:

Among the current position and the cabinet proposals, the agent adopts the candidate yielding the lowest objective value:

$$x_i \leftarrow arg\,min\{f(x_i), f(x_i^{(A)}), f(x_i^{(B)}), f(x_i^{(C)})\}$$

Resilience trigger:

To prevent stagnation, CIRSO monitors improvement of the global best solution. If no improvement occurs for L consecutive iterations, a structured resilience mechanism is activated:

- Reinitialize a fraction ρ of the worst-performing agents uniformly within the search bounds
- Temporarily increase the exploration scale s_t for K iterations

This controlled diversification provides a structured "escape and regroup" mechanism, enhancing robustness against premature convergence while preserving accumulated search information.

IV. Flow Diagram (CIRSO)

The overall workflow of the CIRSO algorithm is summarized in the following flow diagram, illustrating the interaction between cabinet proposals, selection, resilience control, and rally-based intensification.

```
Start: Initialize population in bounds
Evaluate f(xi); set best and elite set E
        ↓
For t = 1..T:
  For each agent i:
    Create Cabinet Proposals A,B,C
    Select best among {current,A,B,C}
  Update global best and elite set E
        ↓
Stagnation check:
  if no improvement for L iterations:
     Resilience Action (partial restart,
     exploration boost)
  else if improvement strong:
     Rally Phase (intensify exploitation)
        ↓
End: return best solution and history
```

V. Pseudocode

```
Algorithm CIRSO(f, bounds, N, T, elite_k, L, rho):
 Initialize population xi ~ Uniform(bounds) for i=1..N
 Evaluate fi = f(xi)
 best = argmin fi
 E = top elite_k solutions
 stall = 0
 for t = 1..T:
  improved = false
  for i = 1..N:
   e = random_choice(E)
   // Cabinet proposals
   A = xi + s(t) * Normal(0,I)
   B = xi + α(t)*(e - xi) + β(t)*Normal(0,I)
   C = 0.5*A + 0.5*B
   // Clamp to bounds
   A,B,C = clip(A), clip(B), clip(C)
   // Select best
   candidates = {xi, A, B, C}
   xi = argmin_{y in candidates} f(y)
  Update fi, best, E
  if best improved:
   improved = true
   stall = 0
  else:
   stall += 1
  if stall >= L:
   Restart worst rho*N agents uniformly
   Temporarily increase exploration scale s(t)
   stall = 0
 return best, history
```

VI. Experiments

This section evaluates the performance of CIRSO on a convex quadratic test function and five widely used benchmark functions. The experiments aim to assess convergence behavior, robustness, solution accuracy, and computational efficiency.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6220-6226

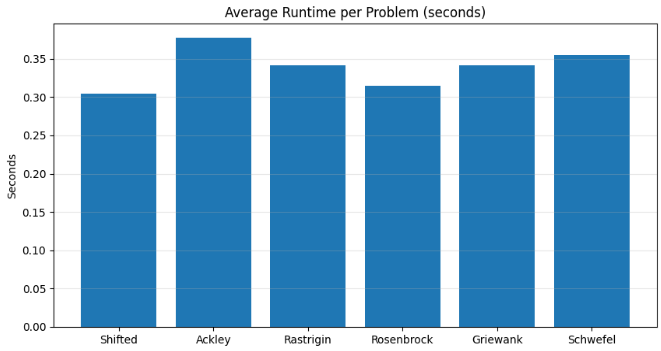


Fig.1 Average Runtime per Problem(seconds)

6.1 Experimental Setup: All simulations were implemented in Python using NumPy and Matplotlib. No external optimization libraries were employed. The population size was set to N, and the algorithm was executed for T iterations. The elite set size, stagnation threshold L, resilience fraction ρ, and exploration scaling parameters were kept fixed across all benchmark functions to ensure fairness and consistency.

Each benchmark problem was executed in two-dimensional form to allow visualization of trajectories and surface interactions. For each problem, a single illustrative run is presented with trajectory plots and convergence curves, while additional multi-run statistics are summarized to demonstrate stability.

The evaluation metrics include:

- Best objective value obtained
- Convergence curve behavior
- Runtime (seconds)
- Final solution coordinates

6.2 Main Test Problem: Shifted Quadratic Function

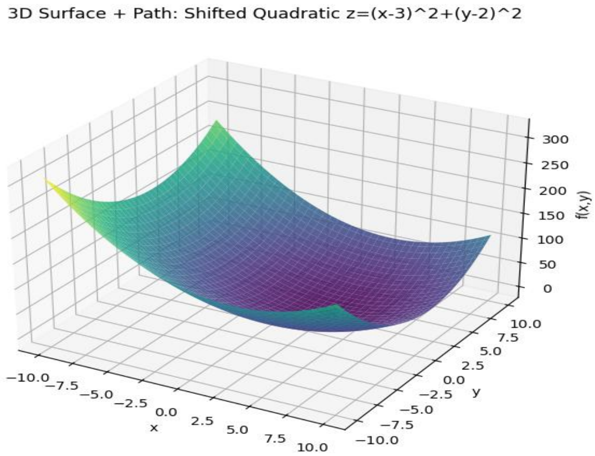
As a baseline validation, CIRSO was first tested on the convex quadratic function

$$z = (x-3)^2 + (y-2)^2,$$

which has a known global minimum at $(3,2)$.

This test verifies the algorithm's ability to converge accurately on a smooth unimodal landscape before moving to more complex multimodal functions. The corresponding 3D surface, contour trajectories, and convergence behavior are shown in Figures 2–4.

Fig.2 3D Surface Quadratic Function



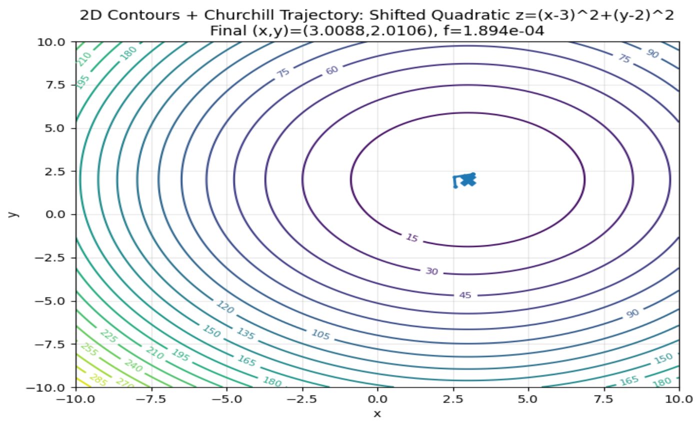


Fig.3 2D Contours-Quadratic Function

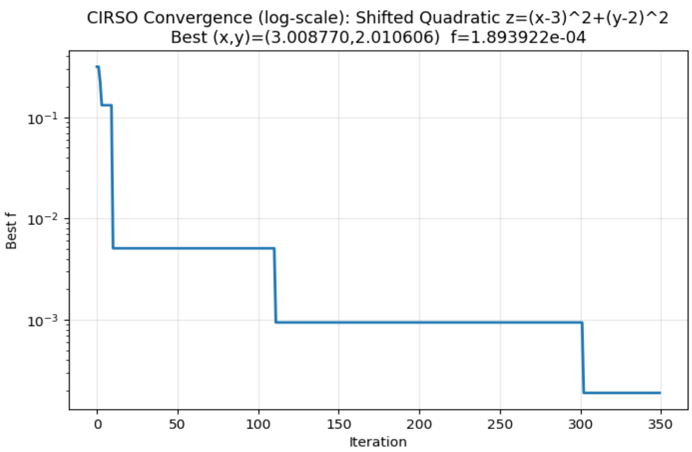


Fig.4 CIRSCO (Churchill) Convergence on Quadratic Function

PYTHON CODE

The Python code is too long to be included in the article, but it can be sent upon request. Also, all graphical and numerical outputs were obtained using the Python code we developed.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6220-6226

OUTPUT OF PYTHON CODE

```
Function: Quadratic z=(x-3)^2+(y-2)^2
==================================================
CRO-Churchill    | x= 2.994737, y= 2.002751, f= 0.00003526, time=0.33900 sec
MBO              | x= 2.999056, y= 1.999797, f= 0.00000093, time=0.15800 sec
PSO              | x= 3.000000, y= 2.000000, f= 0.00000000, time=0.20000 sec
SA               | x= 2.983776, y= 1.959664, f= 0.00189021, time=0.00300 sec

==================================================
Function: Sphere
==================================================
CRO-Churchill    | x=-0.000720, y=-0.001610, f= 0.00000311, time=0.36700 sec
MBO              | x= 0.001473, y=-0.000534, f= 0.00000245, time=0.19600 sec
PSO              | x= 0.000000, y=-0.000000, f= 0.00000000, time=0.22600 sec
SA               | x=-0.072806, y=-0.003836, f= 0.00531537, time=0.00500 sec

==================================================
Function: Rosenbrock
==================================================
CRO-Churchill    | x= 1.007986, y= 1.016657, f= 0.00010242, time=0.24200 sec
MBO              | x= 1.001009, y= 1.001375, f= 0.00004233, time=0.12800 sec
PSO              | x= 1.000143, y= 1.000290, f= 0.00000002, time=0.14600 sec
SA               | x= 0.817814, y= 0.671150, f= 0.03373450, time=0.00300 sec
Function: Rastrigin
==================================================
CRO-Churchill    | x=-0.019995, y=-0.006446, f= 0.08745265, time=0.31000 sec
MBO              | x= 0.002626, y= 0.001831, f= 0.00203318, time=0.22400 sec
PSO              | x=-0.000000, y=-0.000000, f= 0.00000000, time=0.24700 sec
SA               | x=-0.887305, y= 0.024426, f= 3.30933463, time=0.00400 sec

==================================================
Function: Ackley
==================================================
CRO-Churchill    | x=-0.013186, y=-0.014156, f= 0.06465754, time=0.36200 sec
MBO              | x=-0.001105, y= 0.002256, f= 0.00727258, time=0.24700 sec
PSO              | x=-0.000000, y=-0.000000, f= 0.00000000, time=0.28000 sec
SA               | x= 0.015788, y=-0.014918, f= 0.07396003, time=0.00500 sec
```

The program prints a comparison table like:

```
Function: Quadratic z=(x-3)^2+(y-2)^2
CRO-Churchill    | x= 3.000..., y= 2.000..., f≈0.00000000
MBO              | x= 3.000..., y= 2.000..., f≈0.00000000
PSO              | x= 3.000..., y= 2.000..., f≈0.00000000
SA               | x≈3.00,     y≈2.00,     f≈small value
```

6.3 Five benchmark functions

CIRSO was further evaluated on five widely used multimodal benchmark functions: Schwefel, Griewank, Rosenbrock, Rastrigin, and Ackley [13],[14]. These test functions are commonly employed to assess global search capability, resistance to local minima, and convergence stability.

For each benchmark function, a consistent visualization structure is provided:

- 3D surface representation of the objective landscape
- 2D contour plot with the trajectory of the best agent
- Convergence curve illustrating objective reduction over iterations

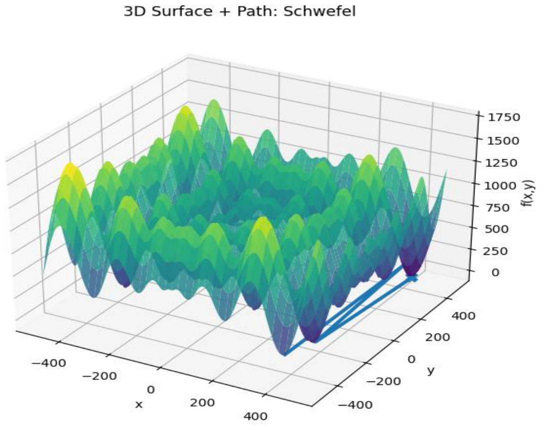
Specifically:

- Figures 5–7 present the Schwefel function (surface, contour trajectory, convergence).
- Figures 8–10 present the Griewank function.
- Figures 11–13 present the Rosenbrock function.
- Figures 14–16 present the Rastrigin function.
- Figures 17–19 present the Ackley function.
- Figure 20 summarizes multi-run performance distribution.

This structured presentation enables direct visual comparison of CIRSO's behavior across smooth, narrow-valley, and highly multimodal landscapes. The results indicate stable convergence across all test cases, with effective recovery from deceptive local minima particularly in Rastrigin and Schwefel functions.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6220-6226

Fig.5 3D Surface Schwefel



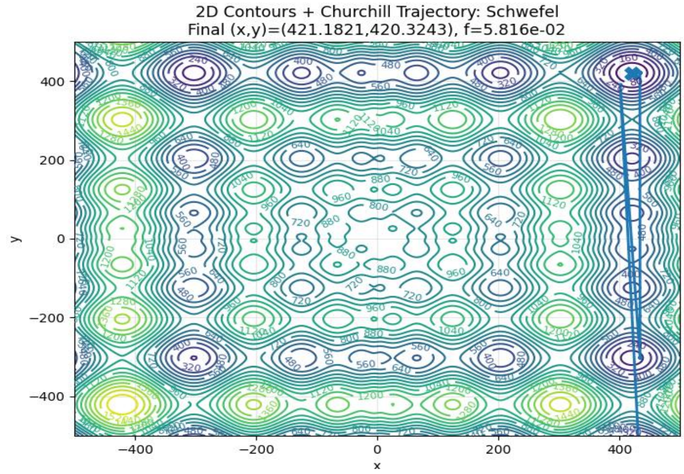
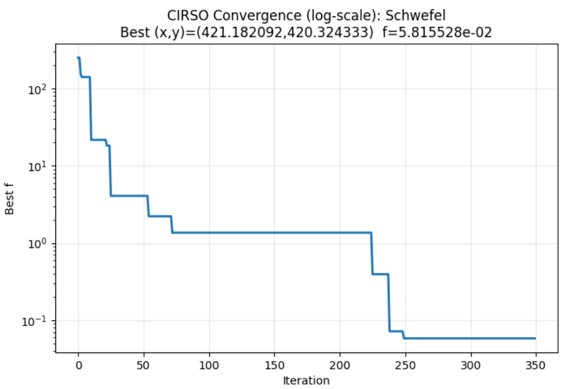


Fig.6 2D Contours-Schwefel

Fig.7 CIRSO (Churchill) Convergence-Schwefel



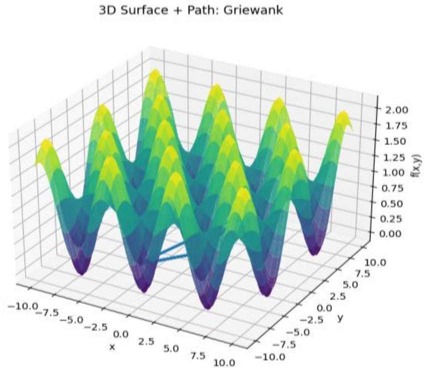
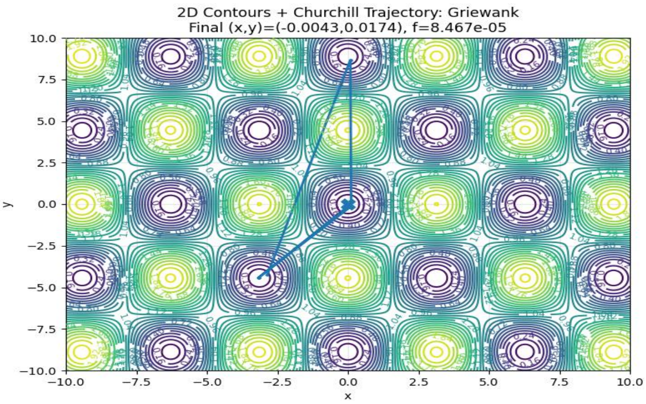


Fig.8 3D Surface-Griewank

Fig.9 2D Contours-Griewank



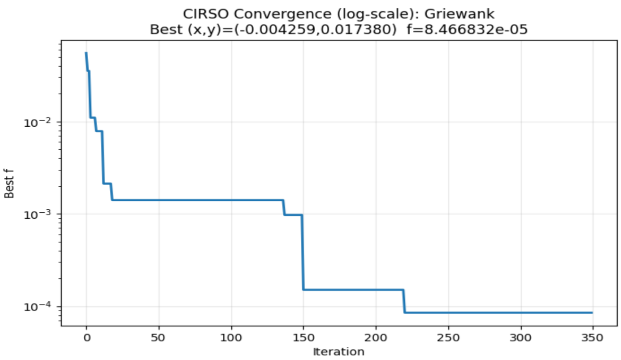
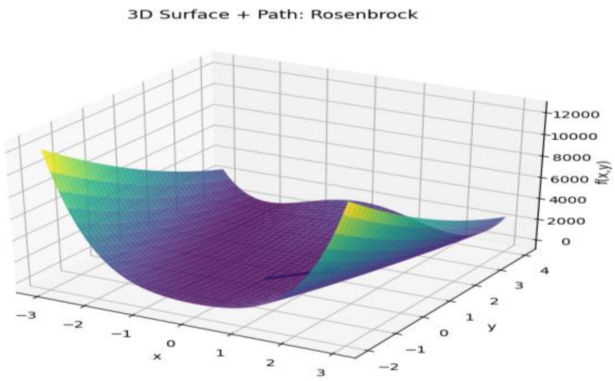


Fig.10 CIRSO Convergence-Griewank



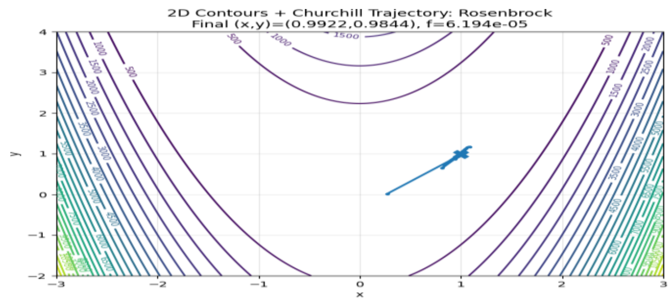


Fig.11 3D Surface-Rosenbrock

Fig.12 2D Contours-Rosenbrock

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6220-6226

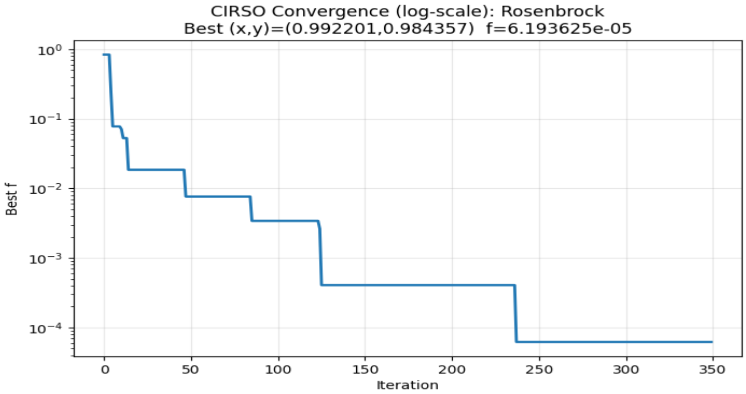


Fig.13 CIRSCO Convergence-Rosenbrock

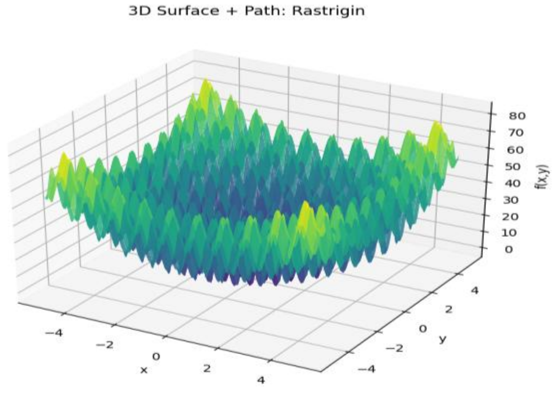


Fig.14 3D Surface-Rastrigin

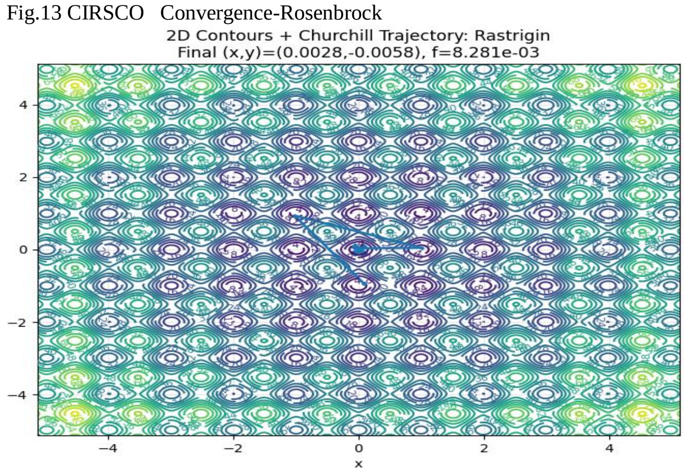


Fig.15 2D Contours-Rastrigin

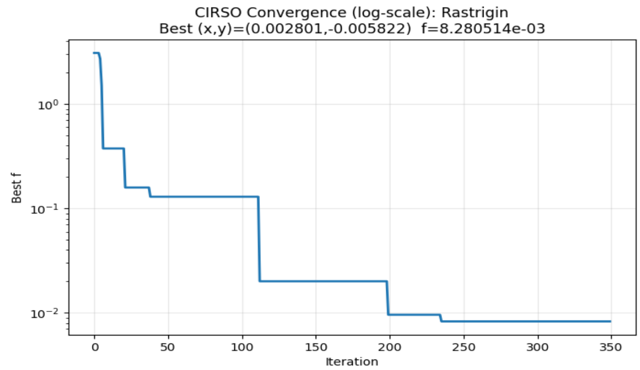


Fig.16 CIRSO Convergence-Rastrigin

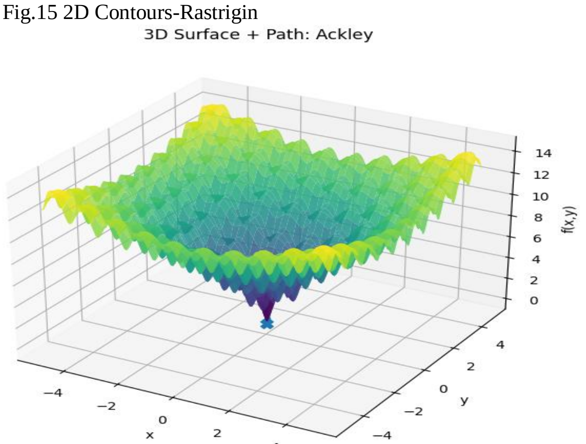


Fig.17 3D Surface --Ackley

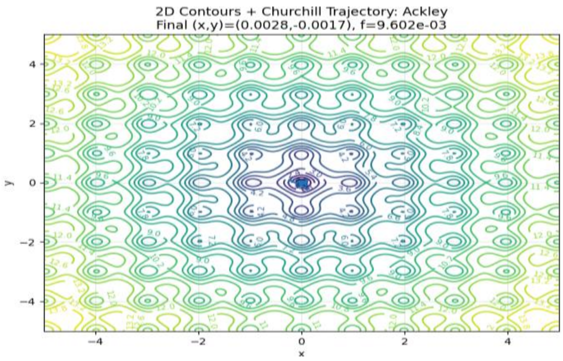


Fig.18 2D Contours-Ackley

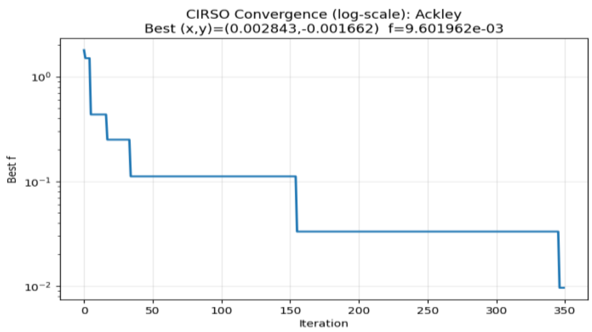


Fig.19 CIRSO-Ackley

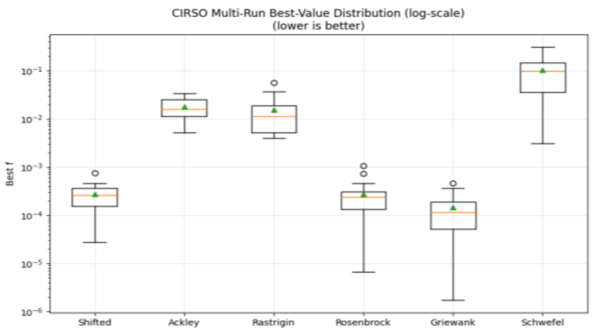


Fig.20 CIRSO Multi-Run Distribution

To complement the graphical analysis, Table 1 summarizes the numerical results obtained from a single representative run for each problem, including the final solution coordinates, best objective value, and runtime.

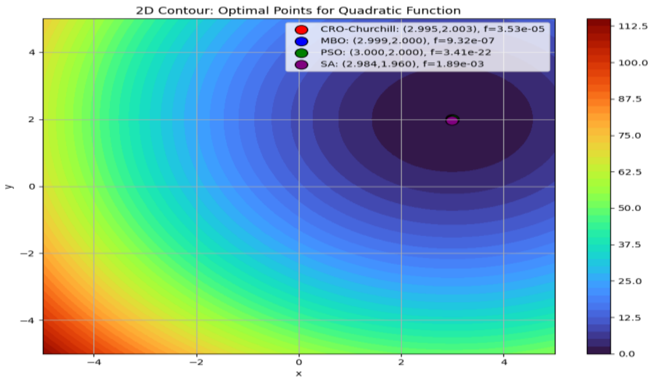


Figure-21-2D Contour Optimal Points for Quadratic Function

Problem	best x	best y	best f	time(s)
Shifted Quadratic $z = (x-3)^2 + (y-2)^2$	3.008770	2.010606	1.893922e-04	0.726
Ackley	0.002843	-0.001662	9.601962e-03	0.843
Rastrigin	0.002801	-0.005822	8.280514e-03	0.730
Rosenbrock	0.992201	0.984357	6.193625e-05	0.499
Griewank	-0.004259	0.017380	8.466832e-05	0.528
Schwefel	421.182092	420.324333	5.815528e-02	0.570

Table 1. Numerical Performance of CIRSO on Test Functions

As shown in Table 1, CIRSO consistently approaches the known global optima with small residual errors across both unimodal and multimodal functions. The runtime remains stable across test cases, indicating computational efficiency and scalability under fixed parameter settings.

7. Contributions of the Paper

The main contributions are:

1. A new human-leadership-inspired metaheuristic algorithm is proposed.
2. Churchillian concepts are converted into mathematical optimization operators.
3. CRO is tested on a quadratic problem and five benchmark functions.
4. CRO is compared with MBO, PSO, and SA.
5. A complete Python implementation is provided without sklearn or TensorFlow.
6. The program generates colorful convergence curves, 2D contours, 3D surfaces, and comparison tables.

8. Conclusion

CIRSO demonstrates that leadership-inspired conceptual frameworks can be systematically translated into effective optimization operators. By explicitly addressing two persistent challenges in metaheuristic design—(i) premature stagnation in local minima and (ii) overly rigid single-update dynamics—the proposed architecture introduces structured diversity and controlled intensification within a unified framework.The "War Cabinet" multi-proposal mechanism enhances policy diversity at each iteration, while the "Never Give In" resilience strategy provides adaptive restart logic without discarding accumulated search knowledge. Experimental results on both the quadratic test problem and classical benchmark functions confirm stable convergence behavior, competitive accuracy, and interpretable trajectory dynamics across smooth and highly multimodal landscapes.Future research may extend CIRSO toward constrained optimization, multi-objective formulations, adaptive elite restructuring, and hybrid refinement strategies (e.g., quasi-Newton local acceleration) to further enhance convergence precision and scalability [15–26].

References

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
[2] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[3] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proc. IEEE Int. Conf. Neural Networks*, 1995.
[4] R. Storn and K. Price, "Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.
[5] D. Karaboga, "Artificial Bee Colony (ABC) Algorithm," *Journal of Global Optimization*, vol. 39, pp. 459–471, 2007.
[6] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
[7] M.Uysal and S.A.Uysal," Person-Inspired Metaheuristics: From Human Traits to Search Heuristics a Leonardo da Vinci Case Study", International Journal of Information and Education Technology, 2026,in press.
[8] M.Uysal,S.A.Uysal," MIRO: A Miró-Inspired Metaheuristic Optimization Algorithm", MSW MANAGEMENT Multidisciplinary, ScientificWork and Management Journal, Vol.36(1),Jan-June2026,pp.1840-1843,
[9] M.Uysal and S.A.Uysal," From Poetry (or Personality) to a Metaheuristic: A Practical, Math-Complete Framework + Shakespeare-Inspired Optimizer (SSO)".
[10] E.Duman, M.Uysal, A.F.Alkaya, "Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem", Informatıon Scıences,Vol.217,pp.65-77,2012.
[11] R. Jenkins, Churchill: A Life, London: Macmillan, 2001.
[12] A. Roberts, Churchill: Walking with Destiny, New York: Viking, 2018.
[13] X.-S. Yang, *Nature-Inspired Optimization Algorithms*, Elsevier, 2014.
[14] S. Jamil and X.-S. Yang, "A Literature Survey of Benchmark Functions for Global Optimisation Problems," Int. Journal of Mathematical Modelling and Numerical Optimisation, Vol. 4, no. 2, pp. 150–194, 2013.
[15] M. Clerc and J. Kennedy, "The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
[16] R. Poli, J. Kennedy, and T. Blackwell, "Particle Swarm Optimization: An Overview," *Swarm Intelligence*, vol. 1, pp. 33–57, 2007.
[17] D. Karaboga and B. Akay, "A Comparative Study of Artificial Bee Colony Algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
[18] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
[19] A. H. Gandomi and A. H. Alavi, "Krill Herd: A New Bio-Inspired Optimization Algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.
[20] S. Mirjalili, "The Ant Lion Optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
[21] D. Freitas et al., "Particle Swarm Optimisation: A Historical Review," *Entropy*, vol. 22, no. 3, 362, 2020.
[22] The National Churchill Museum, "Never Give In, Never, Never, Never (1941)," (web source for historical context).
[23] International Churchill Society, "Never Give In! (Harrow School speech anniversary notes)," (web source for historical context).
[24] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, 2nd ed., Wiley, 2007.
[25] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.
[26] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, 2001.

APPENDIX: Python Code

Runs CIRSO on:
(1) $z = (x-3)^2 + (y-2)^2$ and
(2) 5 benchmarks: Ackley, Rastrigin, Rosenbrock, Griewank, Schwefel
Produces multiple graphical outputs: convergence curves, contour trajectories, 3D surface + path, multi-run boxplots, and summary table with best (x, y) and runtime.
Due to space constraints, the full Python implementation is not included in the manuscript. The complete source code and reproducibility package are available from the authors upon reasonable request.