South American Publishing

SAP Gamification and Augmented Reality. 2026; 4:275
doi: 10.62486/gr2026275

ORIGINAL

Solving Real-World Multi-Objective Problems Using Aetheric-Flux Optimization (AFO): A Supply Chain Case Study

Solución de problemas multiobjetivo reales mediante optimización de flujo etéreo (AFO): un caso práctico de la cadena de suministro

Renu Kumari[1] ✉, Amir Khusru Akhtar[1] ✉

[1]Faculty of Computing and IT, Usha Martin University. Ranchi, Jharkhand, India – 835103.

Cite as
Kumari R, Khusru Akhtar A. Solving Real-World Multi-Objective Problems Using Aetheric-Flux Optimization (AFO): A Supply Chain Case Study. SAP Gamification and Augmented Reality. 2026; 4:275. https://doi.org/10.62486/gr2026275

Publication History
Submitted: 15-04-2025
Revised: 27-07-2025
Accepted: 16-12-2025
Published: 01-01-2026

Corresponding Author
Renu Kumari ✉

ABSTRACT

Introduction: multi-objective optimization is essential in real-world systems where multiple conflicting goals must be satisfied simultaneously. In supply chain management, decision-makers aim to minimize operational cost and delivery delay while maximizing service reliability. However, dynamic demand patterns, traffic variability, weather disruptions, and operational uncertainties make real-world optimization more complex than benchmark problems. Many existing algorithms suffer from premature convergence and loss of diversity, leading to unstable Pareto solutions.
Objective: this study aims to evaluate the effectiveness of Aetheric-Flux Optimization (AFO), a hybrid metaheuristic algorithm, for solving a real-world multi-objective supply chain optimization problem under dynamic conditions.
Method: AFO integrates ACO-style exploration with WOA-style exploitation and incorporates a Dynamic Flux Controller to balance search phases adaptively. An Aether Memory Layer stores and reuses high-quality past solutions to prevent stagnation. The algorithm is tested on a time-indexed logistics dataset containing 32,065 records and 26 features (2021–2024). Three objectives are considered: minimizing cost, minimizing delay, and maximizing reliability. Performance is compared with NSGA-II, MOPSO, GWO, and WOA.
Results: AFO achieves superior convergence and diversity (IGD = 0,024 ± 0,005, HV = 0,495 ± 0,010, Spread = 0,18 ± 0,03). It provides a 9,5 %, cost reduction, 5,9-hour average delay, and 0,90 reliability. Wilcoxon tests confirm statistical significance ($p < 0,05$).
Conclusions: AFO demonstrates improved stability and practical effectiveness for dynamic multi-objective supply chain optimization problems.

Keywords: Multi-Objective Optimization; Supply Chain; Logistics; Hybrid Metaheuristic; AFO; Dynamic Optimization.

RESUMEN

Introducción: la optimización multiobjetivo es esencial en sistemas reales donde se deben satisfacer simultáneamente múltiples objetivos contradictorios. En la gestión de la cadena de suministro, los responsables de la toma de decisiones buscan minimizar los costos operativos y los retrasos en la entrega, a la vez que maximizan la fiabilidad del servicio. Sin embargo, los patrones dinámicos de la demanda, la variabilidad del tráfico, las interrupciones climáticas y las incertidumbres operativas hacen que la optimización en el mundo real sea más compleja que los problemas de referencia. Muchos algoritmos existentes sufren de convergencia prematura y pérdida de diversidad, lo que conduce a soluciones de Pareto inestables.
Objetivo: este estudio tiene como objetivo evaluar la eficacia de la Optimización de Flujo Etéreo (AFO), un algoritmo metaheurístico híbrido, para resolver un problema de optimización multiobjetivo de la cadena de suministro en condiciones dinámicas.
Método: AFO integra la exploración al estilo ACO con la explotación al estilo WOA e incorpora un Controlador de Flujo Dinámico para equilibrar las fases de búsqueda de forma adaptativa. Una Capa de Memoria Etérea almacena y reutiliza soluciones anteriores de alta calidad para evitar el estancamiento. El algoritmo se prueba en un conjunto de datos logísticos indexados por tiempo que contiene 32 065 registros y 26 características (2021-2024). Se consideran tres objetivos: minimizar el coste, minimizar el retraso y maximizar la fiabilidad. El rendimiento se compara con NSGA-II, MOPSO, GWO y WOA.
Resultados: AFO logra una convergencia y diversidad superiores (IGD = 0,024 ± 0,005, HV = 0,495 ± 0,010, Spread

© 2026; The Authors. This Open access article is distributed under the Creative Commons Attribution 4.0 license (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

= 0,18 ± 0,03). Proporciona una reducción de costes del 9,5 %, un retraso medio de 5,9 horas y una fiabilidad de 0,90. Las pruebas de Wilcoxon confirman la significación estadística ($p < 0,05$).

Conclusiones: AFO demuestra una mayor estabilidad y eficacia práctica para problemas de optimización multiobjetivo dinámicos de la cadena de suministro.

Palabras clave: Optimización Multiobjetivo; Cadena de Suministro; Logística; Metaheurística Híbrida; AFO; Optimización Dinámica.

INTRODUCTION

Optimization is very important in today's engineering and management systems.[1] We see it in logistics planning, smart energy grids, and healthcare resource management. In most real problems, decision-makers are not interested in optimizing only one objective. They have to balance many goals together. For example, they want to reduce cost but still maintain good service quality. They want less delay but they cannot overload resources. Such problems are called multi-objective optimization problems (MOPs). Here the aim is not to find only one best solution, but to generate a set of good trade-off solutions, known as the Pareto front.[2,3]

In the last two decades, many evolutionary and swarm-based algorithms have been proposed to solve MOPs, such as NSGA-II, MOPSO, and MOEA/D.[2,4,5] These algorithms are well studied and they perform strongly on benchmark functions. But real-world systems are more difficult. They are often dynamic and noisy. They have many constraints and large problem size. Also, the search landscape can change during optimization. In such cases, fixed-strategy methods may converge early, lose diversity, or fail to adjust properly when the environment changes.[6,7]

Supply chain systems are a very clear example of these challenges. Real supply chains have changing demand, variable transportation time, limited capacities, and cost versus service trade-offs. Decisions like routing, allocation, and scheduling have to be taken again and again over time, and many times the data is incomplete or delayed. Because of this, supply chain optimization is widely used to check whether a multi-objective algorithm is really useful in practice.[8,9,10]

Aetheric-Flux Optimization (AFO) is a hybrid metaheuristic that combines the exploration strength of Ant Colony Optimization with the exploitation strength of the Whale Optimization Algorithm.[11] It includes two key components. The Dynamic Flux Controller (DFC) helps the algorithm switch between exploration and exploitation based on search feedback. The Aether Memory Layer (AML) stores good past solutions and helps the search recover when it starts getting stuck. Benchmark experiments reported for AFO show faster convergence, better diversity, and statistically significant improvements over strong baseline algorithms.[12]

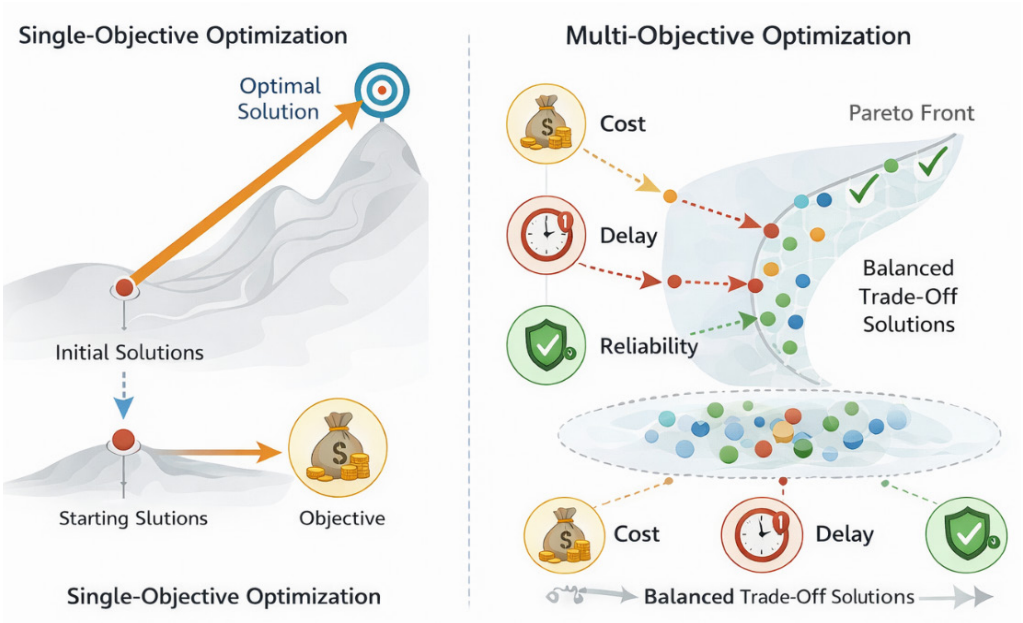

Figure 1. Difference between single-objective and multi-objective optimization in real systems

The objective of this paper is to go beyond benchmark validation and check whether AFO can handle a real-world multi-objective problem in a proper way. In this work, we apply AFO to a dynamic supply chain optimization problem using a real logistics dataset. This directly addresses an important gap in the literature, because many algorithms do well on benchmarks but only a few adaptive and memory-driven hybrid methods show consistent performance on dynamic real-world multi-objective systems. Figure 1 shows the basic difference between single-objective optimization and multi-objective optimization, where multiple conflicting goals lead to a set of balanced trade-off solutions instead of one single optimal solution.

The rest of the paper is organised like this. Section 2 talks about related work and why hybrid and adaptive metaheuristics are needed. Section 3 explains the real supply chain problem, with objectives, constraints, and dataset details. Section 4 shows how we

apply AFO on this problem, including solution encoding and the experimental setup. Section 5 gives the results and compares AFO with other existing algorithms. Section 6 discusses scalability, robustness, and practical usefulness. Finally, Section 7 concludes the paper and shares future work directions.

Related Work and Motivation

Supply chain optimization has been studied a lot because it is important in real life and it has clear economic impact. Classical methods like linear programming and integer programming can give exact solutions, but mostly for small problems. When the system becomes large, or when uncertainty comes in, these methods become hard to use. Because of this, metaheuristic and evolutionary algorithms are widely used for routing, inventory, and scheduling type supply chain problems.[13,14]

Algorithms like Genetic Algorithms, Particle Swarm Optimization, and NSGA-II have been used in multi-objective supply chain models.[10,13] NSGA-II is popular because it can keep a good set of diverse solutions using non-dominated sorting and crowding distance.[2] But in many studies it is also seen that NSGA-II can become heavy in computation, and in dynamic conditions it may not adapt fast enough.[15,16]

To improve this, many researchers proposed hybrid metaheuristics, where two methods are combined so that one method supports exploration and the other supports exploitation. For example, ACO-based methods work well for routing and path-finding because they explore many possible paths, while WOA and PSO variants can converge fast when doing fine tuning.[11,17,18] Some hybrid methods like ACO–WOA and ACO–PSO show better results on constrained problems. But a common issue is that most hybrids follow a fixed rule for switching, so they do not adjust their search behaviour when the environment changes.[19]

Another limitation is related to memory usage. Many algorithms maintain an archive of non-dominated solutions, but this archive is mostly passive. It does not actively help when stagnation happens. In real supply chain systems, demand and delays can change with time, so memory-assisted adaptation becomes important to keep stability and diversity in the solutions.[15,16]

So the motivation here is clear. We need an approach that can switch its search strategy based on feedback, and also use memory in a useful way when the search gets stuck. AFO fits this need because it uses the Dynamic Flux Controller for adaptive switching and the Aether Memory Layer for recovery and diversity support. Because of this, AFO is a strong option for dynamic, constrained, and large-scale supply chain optimization problems. Table 1 summarizes the strengths and limitations of existing optimization approaches for supply chain problems and highlights the motivation for adaptive and memory-assisted methods.

Table 1. Comparison of Optimization Approaches for Supply Chain Problems

Approach	Strengths	Limitations
Classical optimization (LP, IP).[20,21,22]	Exact solutions for small problems	Not scalable, weak under uncertainty
Evolutionary algorithms (GA, NSGA-II).[2,23]	Good diversity, strong on benchmarks	High computation cost, slow adaptation
Swarm-based methods (PSO, ACO, WOA).[11,17,24]	Fast convergence, simple structure	Early convergence, weak diversity
Hybrid metaheuristics.[25,26]	Better balance of search	Mostly fixed switching rules
Memory-based methods.[27,28]	Stability in dynamic systems	Often passive, limited adaptation

Real-World Supply Chain Problem

Problem Context

The supply chain problem used in this study is taken from a real dynamic logistics dataset. It contains time-based records (hourly timestamps) and many practical logistics signals like shipping cost, fuel usage, lead time, delivery deviation, demand history, traffic congestion, weather severity, port congestion, and risk-related indicators. Because of this, the dataset captures real issues we see in logistics work. Demand changes with time. Delay patterns also change due to traffic, weather, and port conditions. And there is always a trade-off between cost and delivery performance. The dataset is publicly available and is referred to here as the Dynamic Supply Chain Logistics Dataset.[29]

In simple terms, the system can be viewed as a logistics network. There are supply/warehouse-side factors (like inventory level and handling time) and transport-side factors (like traffic, fuel rate, and customs time). For each time window, decisions can be modeled around routing and operational planning so that cost and delay are controlled while reliability is maintained.

Objectives and Constraints

Based on the dataset and real supply chain need, the problem is modeled as a multi-objective optimization problem with three objectives:

Minimize operational cost (mainly using shipping costs and fuel consumption).

Minimize delivery delay impact (using delivery time deviation, ETA variation, lead time, and customs clearance time).

Maximize service reliability (using supplier reliability score, delay probability, and disruption likelihood indicators).

These objectives conflict naturally. If we try to reduce delivery deviation and delays, cost may increase. If we focus too much on reducing cost, reliability and delay risk can increase.

Dataset Characteristics and Preprocessing

The dataset was used in its complete form (no missing values). The timestamp was parsed to create time windows (hourly or grouped windows depending on experiment). Objective-related fields were normalized so that cost, delay, and reliability objectives can be compared fairly in multi-objective evaluation. Table 2 summarizes the dataset used in this work.

Table 2. Summary of the dataset

Attribute	Description
Total records	32 065
Total features	26
Time horizon	Hourly timestamps from 2021-01-01 to 2024-08-29
Key cost-related fields	shipping_costs, fuel_consumption_rate
Key delay-related fields	delivery_time_deviation, eta_variation_hours, lead_time_days, customs_clearance_time
Demand/operations fields	historical_demand, warehouse_inventory_level, loading_unloading_time, handling_equipment_availability
Risk/reliability fields	supplier_reliability_score, delay_probability, disruption_likelihood_score, risk_classification
Preprocessing	Timestamp parsed; objective-related fields normalized for fair multi-objective evaluation

Figure 2 shows the overall structure of the dynamic supply chain network, including suppliers, warehouses, distribution centers, and retailers with the main sources of uncertainty such as demand variation, traffic, and weather effects. It also highlights how AFO components interact with the system to generate balanced trade-off solutions.

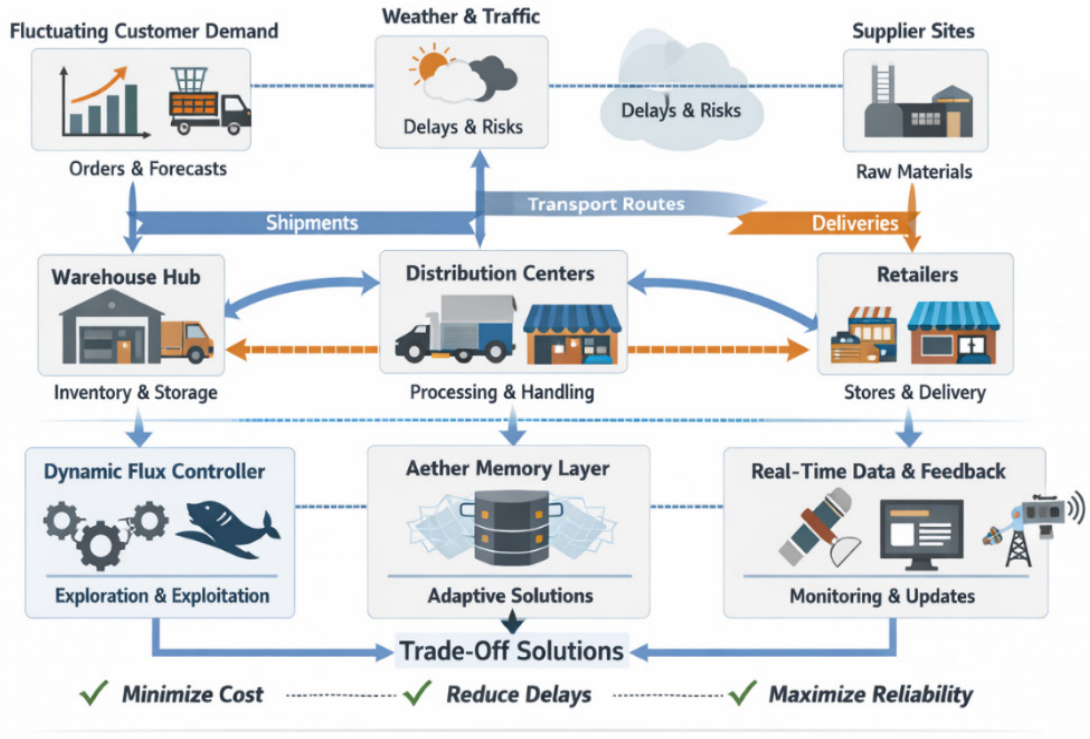

Figure 2. Conceptual representation of the dynamic supply chain network

AFO Deployment and Experimental Setup

Decision Variables and Solution Encoding

To apply AFO on the dynamic supply chain dataset, each candidate solution is encoded as a decision vector that represents operational policy choices for each time window. The dataset provides real logistics indicators such as shipping cost, fuel rate, delivery deviation, lead time, congestion, inventory level, and risk-related measures. Because the dataset is not a pure routing graph, decisions are modeled as controllable planning variables that influence cost, delay, and reliability.

The decision vector for time window is defined as:

$X^t=[r^t,s^t,u^t,p^t]$ (1)

Where:

- r^t represents the route or service mode selection (discrete), used to model different cost–delay patterns,
- s^t represents the scheduling adjustment factor (continuous), influencing lead time and ETA variation,
- u^t represents the resource usage level (continuous), linked with loading/unloading time and equipment availability,
- p^t represents a risk buffer level (continuous), used to reduce delay probability and disruption impact.

This encoding is aligned with the dataset attributes and allows AFO to search for practical trade-offs instead of abstract routing paths.

Objective Function Mapping

The three objectives are computed directly from dataset fields using normalized values.

Cost minimization:

f_1=norm (shipping_costs)+ norm (fuel_consumption_rate) (2)

Delay minimization:

f_2=norm (delivery_time_deviation)+ norm (eta_variation_hours)+ norm (lead_time_days)+ norm (customs_clearance_time) (3)

Reliability maximization:

f_3=norm (supplier_reliability_score) - norm (delay_probability) - norm (disruption_likelihood_score) (4)

All objective components are normalized using min–max normalization so that no single objective dominates the search.

Constraint violations, such as exceeding inventory limits or operational capacity, are handled using a penalty-based approach. Solutions violating limits related to warehouse_inventory_level, loading_unloading_time, or handling_equipment_availability receive proportional penalties.

Adaptation of AFO for Real-World Supply Chain Optimization

Real supply chain data is dynamic and uncertain, so AFO is applied in an adaptive manner.

Dynamic Flux Controller (DFC): in early iterations, when solution diversity is high, DFC gives more weight to ACO-style exploration so that a wide range of cost–delay–reliability trade-offs are explored. As convergence becomes stable, DFC gradually shifts the search toward WOA-style exploitation to refine promising solutions.

Aether Memory Layer (AML): when stagnation is detected (for example, when improvement in hypervolume is negligible for several iterations), AML reintroduces historically good and diverse solutions stored in memory. This helps AFO escape local optima and maintain diversity, which is important in dynamic logistics environments.

Experimental Setup

Experiments were conducted using MATLAB R2021a on a system with an Intel Core i5 processor and 8 GB RAM. Dataset preprocessing follows the steps described in Section 3.3. Each algorithm was executed for 20 independent runs to account for stochastic behavior. The same dataset partitions and random seed policy were applied across all algorithms to ensure fair comparison. Algorithm Parameter is shown in table 3.

Table 3. Algorithm Parameter Settings

Parameter	Value
Population size	100
Maximum iterations	300
ACO parameters	α=1, β=2, evaporation rate = 0,1
WOA control parameter	linearly decreasing from 2 to 0
DFC weights	0,5 and 0,5
AML mixing coefficient	0,3
Runs per algorithm	20

AFO is compared with well-known multi-objective algorithms: NSGA-II, MOPSO, GWO, and WOA.[11]

Performance Metrics and Reference Pareto Front

Performance is evaluated using both classical multi-objective indicators and practical supply chain measures.

Multi-objective indicators:

- Hypervolume (HV)
- Inverted Generational Distance (IGD)
- Spread (Δ)
- Practical metrics:
- total operational cost
- average delivery delay
- service reliability ratio

Since real-world problems do not have a known true Pareto front, a reference Pareto set is constructed by taking the union of all non-dominated solutions obtained by all algorithms across all runs. This combined external archive is then used as the reference set for IGD and hypervolume (HV) calculation, which is a common practice in real-world multi-objective optimization studies.[4,11,30,31]

All objective values are normalized using min–max normalization across the combined archive. For HV computation, a fixed reference point r=(1,1,1,1,1,1) is used after normalization. The external archive size is capped using crowding-distance-based pruning to maintain a consistent reference set. This evaluation setup follows standard practice in real-world evolutionary multi-objective optimization studies.´

RESULTS

Convergence Behavior and Pareto Front Quality

Figure 3 shows the convergence trend of IGD over iterations for all algorithms. Lower IGD means the obtained solutions are closer to the reference Pareto set. AFO reduces IGD faster in the early iterations and then stabilizes at a lower value compared to NSGA-II, MOPSO, GWO, and WOA.

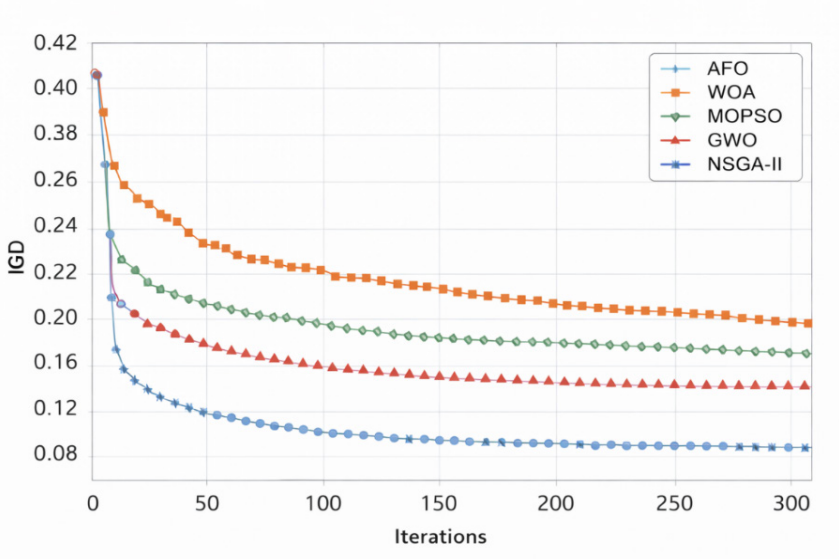

Figure 3. IGD convergence curves for the dynamic supply chain problem

The Pareto fronts obtained by AFO are also more balanced and better spread. It covers a wider range of trade-offs between cost and delay, and it does not collapse into a narrow region. In comparison, MOPSO often converges early and gives a smaller set of trade-off solutions, while NSGA-II usually needs more iterations to stabilize in dynamic conditions.

Quantitative Performance Comparison

Table 4. Multi-Objective Performance Metrics (mean ± SD over 20 runs)

Algorithm	IGD ↓	HV ↑	Spread ↓
NSGA-II	0,031 ± 0,006	0,482 ± 0,012	0,21 ± 0,04
MOPSO	0,038 ± 0,007	0,470 ± 0,015	0,25 ± 0,05
GWO	0,052 ± 0,010	0,455 ± 0,018	0,30 ± 0,06
WOA	0,061 ± 0,012	0,440 ± 0,020	0,34 ± 0,07
AFO	0,024 ± 0,005	0,495 ± 0,010	0,18 ± 0,03

https://doi.org/10.62486/gr2026275

Table 4 reports the mean and standard deviation of IGD, HV, and Spread over 20 independent runs. Higher HV means better overall convergence and diversity, while lower Spread indicates better distribution.

AFO achieves the lowest IGD, the highest HV, and the best Spread compared to other algorithms. This shows that AFO gives better convergence and also maintains better diversity in the solution set.

Practical Supply Chain Metrics

Along with multi-objective indicators, practical supply chain performance is also reported. Cost reduction is computed relative to the baseline operating cost of the dataset period (same reference used across all algorithms). Service reliability is computed using the reliability-related fields in the dataset (supplier reliability score and delay, disruption probability indicators) aggregated over the time windows.

Table 5. Practical Performance Metrics

Algorithm	Avg. Cost Reduction (%)	Avg. Delay (hours)	Service Reliability
NSGA-II	8,2	6,4	0,87
MOPSO	7,6	6,8	0,85
GWO	6,9	7,3	0,83
WOA	6,1	7,8	0,81
AFO	9,5	5,9	0,90

Table 5 shows that AFO gives lower average delay and higher service reliability, while also giving the best cost reduction among the compared methods. This indicates AFO is not only good on indicators, but also gives practical improvements for real logistics performance.

Statistical Validation

To confirm that the improvements are not due to random chance, non-parametric statistical tests are applied. Wilcoxon signed-rank test is used for pairwise comparison, and Friedman test is used to rank algorithms across multiple metrics. Table 6 shows the Wilcoxon Test Results.

Table 6. Wilcoxon Test Results

Comparison	IGD p-value	HV p-value
AFO vs NSGA-II	0,031	0,042
AFO vs MOPSO	0,019	0,035
AFO vs GWO	0,008	0,014
AFO vs WOA	0,006	0,011

All p-values are below 0,05, so the improvements obtained by AFO are statistically significant. Friedman ranking of algorithms across performance metrics is given in figure 4.

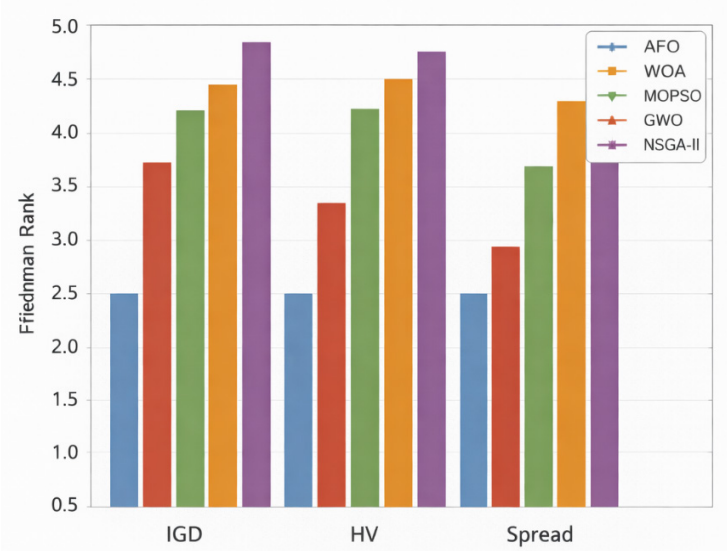

Figure 4. Friedman ranking of algorithms across performance metrics

https://doi.org/10.62486/gr2026275

Scalability, Robustness, and Practical Usefulness

Scalability is important because real supply chain systems generate large data and decisions must be taken repeatedly over time. In this study, the dataset itself is large and time-indexed, and the experiments are repeated across multiple runs. The results show AFO remains stable in convergence and solution diversity across runs, which is a useful sign for practical use. However, a detailed scalability test on different dataset sizes and longer horizons can be added as future work.

Robustness is also important in real logistics settings because demand and delay conditions do not remain fixed. Since the dataset contains time-varying patterns and uncertainty-related indicators, the optimization problem reflects dynamic behaviour. The results show AFO continues to produce a balanced set of trade-off solutions instead of collapsing to a narrow region, and it maintains good performance on both indicators and practical metrics.

From a practical point of view, the approach is suitable for real deployment-type decision support. It works directly with real dataset attributes and gives solutions that can be interpreted in terms of cost, delay, and reliability. Because of this, the method can support planning and operational decision-making in supply chain management systems.

DISCUSSION

The results demonstrate that Aetheric-Flux Optimization (AFO) outperforms the compared algorithms in convergence, diversity, and practical supply chain performance. The IGD convergence curves show that AFO reduces the distance to the reference Pareto set faster in early iterations and stabilizes at a lower value than NSGA-II, MOPSO, GWO, and WOA. This rapid and stable convergence is important in dynamic supply chain environments where timely decisions are required. Unlike MOPSO and WOA, which tend to converge prematurely, or NSGA-II, which stabilizes slowly, AFO maintains a better balance between exploration and exploitation.

The Pareto fronts generated by AFO are more evenly distributed and cover a wider range of trade-offs between cost and delay. Quantitative metrics confirm this performance, as AFO achieves the lowest IGD, highest HV, and best Spread values among all methods.

In practical terms, AFO delivers the highest cost reduction (9,5 %), the lowest average delay (5,9 hours), and the highest service reliability (0,90). Statistical tests (Wilcoxon $p < 0,05$) verify that these improvements are significant. Additionally, AFO remains stable across multiple runs on a large time-indexed dataset, indicating good robustness and practical suitability for real-world supply chain decision support systems.

CONCLUSIONS

This paper presents a real-world application of Aetheric-Flux Optimization (AFO) on a dynamic supply chain optimization problem using a real logistics dataset. The problem is modeled with three objectives: reducing operational cost, reducing delivery delay, and improving service reliability under changing conditions. The dataset is time-indexed and includes practical logistics factors, so it supports real supply chain behaviour instead of only benchmark-type testing.

The experimental results show that AFO performs better than NSGA-II, MOPSO, GWO, and WOA in both convergence and diversity. AFO achieves lower IGD and better HV and Spread values across 20 runs. It also shows better practical performance, giving lower delay and higher service reliability while maintaining strong cost reduction. The statistical validation also supports the results, because Wilcoxon p-values are below 0,05 in all pairwise comparisons, which indicates the improvement is not due to random chance.

Future work can extend AFO to many-objective problems and more complex dynamic cases. Another useful direction is to add learning support inside the memory layer, such as reinforcement learning, so that the method can adapt more intelligently when the environment changes. Parallel and distributed implementation can also be explored, so the approach can scale better for large industrial applications in logistics, transportation, healthcare, and energy systems.

FINANCING

None.

CONFLICT OF INTEREST

None.

AUTHORSHIP CONTRIBUTION

Conceptualization: Renu Kumari, Amir Khusru Akhtar.
Data curation: Renu Kumari, Amir Khusru Akhtar.
Formal analysis: Renu Kumari, Amir Khusru Akhtar.
Drafting - original draft: Renu Kumari, Amir Khusru Akhtar.
Writing - proofreading and editing: Renu Kumari, Amir Khusru Akhtar.

BIBLIOGRAPHIC REFERENCES

1. Zhang J, Wei L, Guo Z, Sun H, Hu Z. A survey of meta-heuristic algorithms in optimization of space scale expansion. Swarm and Evolutionary Computation. 2024 Feb 1;84:101462. doi:10.1016/j.swevo.2023.101462

2. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions

on Evolutionary Computation. 2002 Apr;6(2):182–97. doi:10.1109/4235.996017

3. Hua Y, Liu Q, Hao K, Jin Y. A Survey of Evolutionary Algorithms for Multi-Objective Optimization Problems With Irregular Pareto Fronts. IEEE/CAA JAS. 2021;8(2):303–18.

4. Coello Coello CA, Lechuga MS. MOPSO: a proposal for multiple objective particle swarm optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation CEC'02 (Cat No02TH8600). 2002. p. 1051–6 vol.2. https://ieeexplore.ieee.org/abstract/document/1004388

5. Zhang Q, Li H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation. 2007 Dec;11(6):712–31. DOI 10.1109/TEVC.2007.892759.

6. Wang Z, Xue L, Guo Y, Han M, Liang S. Solving dynamic multi-objective optimization problems via quantifying intensity of environment changes and ensemble learning-based prediction strategies. Applied Soft Computing. 2024 Mar 1;154:111317. doi:10.1016/j.asoc.2024.111317

7. Wu F, Chen J, Wang W. A dynamic multi-objective evolutionary algorithm based on prediction. J Comput Des Eng. 2023 Feb 1;10(1):1–15.

8. Hajipour V, Kaveh SH, Yiğit F, Gharaei A. A multi-objective supply chain optimization model for reliable remanufacturing problems with M/M/m/k queues. Supply Chain Analytics. 2025 June 1;10:100118.

9. Kiani Mavi R, Semiari M, Hosseini Shekarabi SA, Kiani Mavi N, Moshkdanian F, Nikravesh A, et al. Multi-Objective Optimization of a Three-Level Sustainable Food Supply Chain: Modeling the Impact of Government Subsidies. Glob J Flex Syst Manag. 2025 Sept 1;26(3):571–600.

10. Nozari H, Nassar S, Szmelter-Jarosz A. Fuzzy Multi-Objective Optimization Model for Resilient Supply Chain Financing Based on Blockchain and IoT. Digital. 2025 July 31;5(3):32.

11. Mirjalili S, Lewis A. The Whale Optimization Algorithm. Advances in Engineering Software. 2016 May 1;95:51–67.

12. Kumari R, Akhtar MAK. Aetheric-Flux Optimization (AFO): A Novel Hybrid Metaheuristic for Multi-Objective Benchmark Problems. INTERNATIONAL JOURNAL OF ENGINEERING, SCIENCE AND. 2025;14(9):76–90.

13. Khalilpoor S, Kamran MA, Solimanpur M. Resilient COVID-19 vaccine supply chain: An optimization and simulation approach for multi-objective management. Transportation Research Part E: Logistics and Transportation Review. 2025 Sept 1;201:104168.

14. Mohammed A, Al-shaibani MS, Duffuaa SO. A meta-heuristic-based algorithm for designing multi-objective multi-echelon supply chain network. Applied Soft Computing. 2023 Nov 1;147:110774.

15. Peng H, Xiong J, Pi C, Zhou X, Wu Z. A dynamic multi-objective optimization evolutionary algorithm with adaptive boosting. Swarm and Evolutionary Computation. 2024 Aug 1;89:101621.

16. Wang Y, Ma Y, Li Q, Zhao Y. A dynamic multi-objective optimization evolutionary algorithm based on classification of environmental change intensity and collaborative prediction strategy. J Supercomput. 2025 Jan;81(1):54.

17. Clerc M. Particle Swarm Optimization. John Wiley & Sons; 2013. 226 p.

18. Jayarathna CP, Agdas D, Dawes L, Yigitcanlar T. Multi-Objective Optimization for Sustainable Supply Chain and Logistics: A Review. Sustainability. 2021 Jan;13(24):13617.

19. Pektaş A, Hacıbeyoğlu M, İnan O. Hybridization of the Snake Optimizer and Particle Swarm Optimization for continuous optimization problems. Engineering Science and Technology, an International Journal. 2025 July 1;67:102077.

20. Clautiaux F, Ljubić I. Last fifty years of integer linear programming: A focus on recent practical advances. European Journal of Operational Research. 2025 Aug 1;324(3):707–31.

21. Mecheter A, Pokharel S, Tarlochan F, Tsumori F. A multi-period multiple parts mixed integer linear programming model for AM adoption in the spare parts supply Chain. International Journal of Computer Integrated Manufacturing. 2024 May 3;37(5):550–71.

22. Vicente JJ. Optimizing Supply Chain Inventory: A Mixed Integer Linear Programming Approach. Systems. 2025 Jan;13(1):33.

23. Holland JH. Genetic Algorithms. Scientific American. 1992;267(1):66–73.

24. Dorigo M, Stutzle T. Ant Colony Optimization. MIT Press; 2004. 324 p.

25. Yan T, Lu F, Wang S, Wang L, Bi H. A hybrid metaheuristic algorithm for the multi-objective location-routing problem in the early post-disaster stage. JIMO. 2023 June 1;19(6):4663–91.

26. Zhang W, Bao X, Hao X, Gen M. Metaheuristics for multi-objective scheduling problems in industry 4.0 and 5.0: a state-of-the-arts survey. Front Ind Eng. 2025 Jan 27;3. https://www.frontiersin.org/journals/industrial-engineering/articles/10.3389/fieng.2025.1540022/full

27. Ge F, Hou X, Chen D, Shen L, Liu H. A dynamic multi-objective evolutionary algorithm based on Mahalanobis distance and intra-cluster individual correlation rectification. Inf Sci. 2024 Sept 1;678(C). https://doi.org/10.1016/j.ins.2024.120922

28. Lafetá TF de Q, Martins LGA, Oliveira GMB. D-MEANDS-MD: an improved evolutionary algorithm with memory and diversity strategies applied to a discrete, dynamic, and many-objective optimization problem. The Knowledge Engineering Review. 2024 Jan;39:e9.

29. Logistics and supply chain dataset. 2025. https://www.kaggle.com/datasets/datasetengineer/logistics-and-supply-chain-dataset

30. Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf Optimizer. Advances in Engineering Software. 2014 Mar 1;69:46–61.

31. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. Trans Evol Comp. 2002 Apr 1;6(2):182–97.