International Journal of Modern Computer Science and IT Innovations (IJMCSIT)
eISSN: 3087-4289
Volume03 Issue08 | Page No: - 48-54

Received: 30 May 2026, Revised: 24 June 2026, Accepted: 22 July 2026, Published: 20 August 2026

DOI: https://doi.org/10.55640/ijmcsit-v03i08-06

An Adaptive AI Multi-Agent Model for Optimizing Real-Time Data Streaming and System Resilience

Chinedu Okafor [1], Amara Eze [2]

[1] Department of Artificial Intelligence, Lagos Institute of Computing Technology Nigeria

[2] Department of Computer Science and AI, Abuja Institute of Digital Engineering Nigeria

ABSTRACT: Real-time data-streaming environments increasingly require adaptive decision mechanisms capable of coordinating heterogeneous computational resources while maintaining throughput, resilience, and service continuity under dynamic workloads. This paper proposes a conceptual adaptive artificial intelligence (AI) multi-agent model for optimizing real-time data streaming and system resilience. The proposed model combines autonomous agents for stream monitoring, workload allocation, resource coordination, anomaly response, and resilience management within a decentralized decision architecture. Its theoretical foundation is informed by research on distributed allocation, fairness, efficiency, optimization, and computational complexity in multi-agent decision environments. In particular, studies of fair and efficient allocation provide useful principles for balancing competing resource demands, while work on Nash social welfare and allocation algorithms demonstrates the value of optimization objectives that consider collective system utility. The proposed architecture extends these principles from indivisible-resource allocation toward dynamic streaming-resource management. The methodology defines agent roles, state representation, utility functions, adaptive allocation policies, coordination mechanisms, resilience procedures, and evaluation criteria. Analytical findings indicate that adaptive multi-agent coordination can improve resource utilization, reduce the impact of localized failures, and support scalable stream processing when compared conceptually with rigid centralized allocation. The model is particularly relevant to event-streaming environments in which workload intensity, resource availability, and service conditions change continuously. The paper further identifies limitations concerning coordination overhead, convergence, observability, and the absence of empirical benchmarking in the present conceptual study. The framework therefore provides a research foundation for implementing resilient AI-driven streaming systems and for future experimental validation.

KEYWORDS: Adaptive AI, Multi-Agent Systems, Real-Time Data Streaming, Resilience, Resource Allocation, Stream Processing, Distributed Intelligence, Dynamic Optimization, Scalability.

1. Introduction

Real-time data-streaming systems operate under conditions fundamentally different from static computational workloads. Data arrival rates may fluctuate rapidly, processing resources may become temporarily constrained, and individual processing components may fail while the overall service is expected to remain operational. These characteristics create a continuous resource-allocation problem in which computational capacity, network bandwidth, processing priority, and workload placement must be coordinated dynamically. A static allocation policy is therefore insufficient when the system state changes faster than centralized planning cycles can respond.

The theoretical challenge can be interpreted as a dynamic allocation problem involving multiple competing agents and resources. Research on fair division demonstrates that allocation mechanisms must account simultaneously for efficiency, fairness, and the structural constraints of available resources (Bliem, Bredereck, & Niedermeier, 2016). Similarly, the study of fair and efficient allocations illustrates the importance of designing mechanisms that avoid optimizing one criterion while substantially degrading another (Barman, Krishnamurthy, & Vaish, 2018a). Although these studies primarily address allocation of indivisible goods, their underlying optimization principles are relevant to distributed streaming systems where agents compete for limited processing and communication resources.

Recent work on AI-based multi-agent optimization for event streaming further motivates the integration of autonomous coordination, adaptive resource

management, and resilience into a unified architecture (R. Reddy, P. Udayaraju, K. K. Goyal, S. C. R. Vudem, R. Sayana and V. Gummadi, 2026). Such a perspective is particularly important because resilience cannot be treated independently from allocation. If an agent receives a computational assignment that maximizes immediate throughput but creates a single point of failure or excessive resource contention, the apparent optimization may reduce system-level performance.

The objective of this paper is to develop a conceptual adaptive AI multi-agent model that continuously observes streaming conditions, estimates system utility, allocates resources among competing processing agents, and dynamically responds to disruptions. The proposed framework focuses on five interrelated objectives: maintaining real-time processing performance, improving resource utilization, preventing allocation imbalance, detecting degradation or failure, and maintaining service continuity through adaptive reallocation.

The significance of the proposed model lies in its combination of allocation theory and resilience-oriented stream management. Rather than assuming that a single centralized optimizer can maintain an accurate global representation of a rapidly changing streaming environment, the framework distributes decision-making among specialized agents while retaining system-level coordination objectives.

2. Literature Review

Research on allocation provides an important theoretical foundation for adaptive multi-agent resource management. Bliem, Bredereck, and Niedermeier (2016) examine the computational complexity of efficient and envy-free resource allocation under restrictions involving agents, resources, and utility levels. Their work demonstrates that allocation problems can become computationally difficult as the number and characteristics of participating entities increase. This observation directly informs the proposed streaming model: an adaptive architecture must not only determine an efficient allocation but also ensure that its decision process remains computationally feasible as the number of streams and processing agents grows.

Barman, Krishnamurthy, and Vaish (2018a) investigate fair and efficient allocation, while their related work on greedy algorithms examines mechanisms for maximizing Nash social welfare (Barman, Krishnamurthy, & Vaish, 2018b). The significance of these contributions for streaming environments is methodological rather than domain-specific. A streaming platform can similarly define utility according to throughput, latency, processing capacity, reliability, or service priority. Instead of maximizing one local metric, a multi-agent optimizer can seek a collective utility objective that discourages severe resource imbalance.

Caragiannis et al. (2016) analyze the relationship between maximum Nash welfare and fairness. Their work is especially relevant to the proposed model because system-wide optimization may otherwise favor high-capacity agents while neglecting weaker or resource-constrained processing components. In a streaming architecture, such imbalance can produce queue accumulation, latency spikes, and cascading overload. Consequently, a welfare-oriented objective can be interpreted as a mechanism for preventing resource starvation rather than merely as an abstract fairness criterion.

Cole and Gkatzelis (2015) address approximation of Nash social welfare for indivisible items. Their contribution highlights the practical importance of approximation when exact optimization becomes computationally expensive. This principle is applicable to adaptive streaming because real-time systems cannot indefinitely delay decisions while solving a computationally complex optimization problem. Approximate or heuristic allocation policies may therefore be preferable when they provide sufficiently high-quality decisions within strict latency constraints.

Budish (2011) similarly examines approximate competitive equilibrium from equal incomes in the combinatorial assignment problem. The broader implication is that approximate mechanisms can provide useful allocations despite complex combinatorial structures. This supports the design of bounded-computation allocation policies for streaming environments where decisions must be produced continuously.

Amanatidis et al. (2020) investigate maximum Nash welfare and EFX-related allocation questions, while Amanatidis et al. (2023) review progress and open questions concerning fair division of indivisible goods. Together, these studies indicate that allocation theory remains an evolving field involving trade-offs among efficiency, fairness, and computational tractability. For adaptive streaming, these trade-offs translate into balancing throughput, latency, resilience, and resource equality rather than optimizing any single metric.

Chakraborty et al. (2020) study weighted envy-freeness in indivisible item allocation. The concept of weighting is potentially valuable for heterogeneous streaming workloads because different streams may have different priorities, deadlines, or service-level requirements. A resilience-critical stream should not necessarily receive identical treatment to a low-priority analytical stream.

The literature also establishes the complexity of allocation when preferences, resources, and constraints become heterogeneous. Bredereck et al. (2019) investigate high-multiplicity fair allocation using integer-programming techniques, while de Keijzer et al. (2009) analyze complexity associated with efficiency and envy-freeness under additive preferences. These studies reinforce the need for scalable optimization strategies.

The proposed research gap emerges from the difference between allocation-oriented theoretical models and continuously changing real-time streaming environments. The supplied literature provides strong foundations for fairness, welfare maximization, efficiency, and computational tractability, but does not directly provide an integrated adaptive architecture for AI-controlled event-stream processing and resilience. The proposed model positions itself at this intersection.

3. Methodology

3.1 Proposed Multi-Agent Architecture

The proposed architecture consists of a collection of autonomous but coordinated agents operating above a distributed streaming infrastructure. Each agent has a specialized responsibility while sharing selected system-state information with neighboring agents.

The Stream Monitoring Agent observes incoming event rates, queue lengths, processing latency, packet loss, and resource utilization. The Resource Allocation Agent determines the distribution of CPU, memory, bandwidth, and processing capacity among competing workloads. The Workload Prediction Agent estimates near-future demand based on recent stream behavior. The Resilience Agent detects abnormal degradation and initiates failover or workload migration. Finally, the Coordination Agent aggregates local decisions and resolves conflicts when multiple agents attempt to acquire the same constrained resource.

This distributed structure follows the general logic of multi-agent allocation: individual agents possess partial operational knowledge while coordination mechanisms attempt to maximize collective utility. The allocation problem is therefore represented as a dynamic optimization process rather than as a single fixed assignment.

3.2 State Representation

At time (t), the system state can be represented as:

[

S_t = {D_t, Q_t, L_t, R_t, F_t, P_t}

]

where (D_t) represents data-arrival intensity, (Q_t) represents queue conditions, (L_t) represents latency, (R_t) represents available computational resources, (F_t) represents detected failures or degradation, and (P_t) represents workload priorities.

Each agent receives either a local or partially aggregated state representation. The use of distributed observations reduces dependence on a single monitoring component and potentially improves resilience to localized information failures.

3.3 Adaptive Utility Function

The allocation policy is designed around a composite utility function:

[

U = \alpha T - \beta L + \gamma R_u + \delta F_r - \lambda C

]

where (T) denotes throughput, (L) latency, (R_u) resource utilization, (F_r) resilience or fault-recovery capability, and (C) coordination cost. The coefficients determine the relative importance of each objective.

The inclusion of multiple terms is essential. Maximizing throughput alone could lead to excessive resource concentration, while minimizing latency alone could create inefficient resource allocation. Allocation research demonstrates the importance of considering multiple objectives and the consequences of inefficient distribution (Barman, Krishnamurthy, & Vaish, 2018a; Caragiannis et al., 2016).

3.4 Dynamic Resource Allocation

When the monitoring agent detects increased demand, the allocation agent evaluates available capacity and reallocates resources according to predicted workload. For example, suppose Stream A experiences a sudden increase from 20,000 to 60,000 events per second while Stream B remains stable. A static allocation mechanism may allow Stream A's queue to grow until manual intervention occurs. Under the proposed architecture, the prediction and monitoring agents identify the change, after which the allocation agent temporarily increases Stream A's processing capacity.

The policy must also prevent excessive resource capture by a single stream. This is where welfare-oriented allocation becomes useful. Nash-welfare-based approaches provide a theoretical basis for balancing aggregate utility rather than maximizing the performance of only the strongest participant (Barman, Krishnamurthy, & Vaish, 2018b; Cole & Gkatzelis, 2015).

3.5 Resilience Mechanism

Resilience is implemented as a continuous feedback loop:

[

Observe \rightarrow Detect \rightarrow Evaluate \rightarrow Reallocate \rightarrow Recover \rightarrow Verify

]

The system first detects performance degradation or component failure. It then estimates the severity and determines whether local recovery is sufficient. If not, the affected workload is reassigned to an alternative processing agent. The verification stage evaluates whether latency, throughput, and queue conditions return to acceptable levels.

This mechanism differs from conventional static failover because the response is coupled with adaptive resource allocation. A failed processing node does not simply trigger a predefined backup; the system evaluates current capacity before selecting a recovery target.

3.6 Coordination and Approximate Optimization

Exact global optimization may be inappropriate for strict real-time environments because computational complexity increases with the number of agents, resources, and constraints. Research on allocation complexity supports this concern (Bliem, Bredereck, & Niedermeier, 2016; de Keijzer, Bouveret, Klos, & Zhang, 2009).

Accordingly, the proposed model uses a hierarchical approximation strategy. Local agents first perform rapid decisions using current state information. The coordination agent subsequently evaluates system-level conflicts. Approximate optimization is justified because allocation research demonstrates that useful solutions can be obtained without requiring exact optimization of every combinatorial allocation problem (Budish, 2011; Cole & Gkatzelis, 2015).

3.7 Evaluation Framework

Because the present paper develops a conceptual model rather than reporting a completed experimental deployment, evaluation is defined through measurable system indicators rather than fabricated experimental values. The principal indicators are average latency, tail latency, throughput, resource utilization, recovery time, workload loss, allocation imbalance, and coordination overhead.

A representative experimental scenario would compare three configurations: static allocation, centralized adaptive allocation, and the proposed multi-agent adaptive allocation. Workloads would include stable streams, bursty streams, heterogeneous priority streams, and failure-injection scenarios. The principal hypothesis is that the proposed architecture will provide stronger adaptation and recovery characteristics under workload volatility and localized failures.

The framework is consistent with the broader motivation for AI-based multi-agent optimization in event-streaming environments (R. Reddy, P. Udayaraju, K. K. Goyal, S. C. R. Vudem, R. Sayana and V. Gummadi, 2026).

4. Results

The analytical findings of the proposed framework indicate that adaptive multi-agent coordination can address several weaknesses inherent in static resource-allocation approaches. First, separating monitoring, prediction, allocation, and resilience functions allows the system to react to changing conditions without requiring complete architectural reconfiguration. The principal expected benefit is improved responsiveness during workload bursts because resource decisions are linked directly to observed changes in stream conditions.

Second, the model provides a stronger basis for balancing competing workloads. Allocation mechanisms inspired by welfare maximization discourage excessive concentration of resources on a small subset of workloads. This is particularly relevant when several streams simultaneously compete for constrained computational capacity. The literature on Nash welfare demonstrates why collective utility can provide a more balanced objective than isolated utility maximization (Caragiannis et al., 2016; Barman, Krishnamurthy, & Vaish, 2018b).

Third, resilience becomes an optimization objective rather than an independent emergency mechanism. When a processing agent fails, the system evaluates alternative capacity and reallocates workloads according to current conditions. This design reduces dependence on fixed backup assignments and should be particularly effective when failure occurs during a workload burst.

Fourth, the model explicitly recognizes computational overhead as part of optimization. Exact allocation can become difficult as the number of agents and resources increases, as indicated by the complexity-oriented literature (Bliem, Bredereck, & Niedermeier, 2016). Consequently, approximate decisions provide a practical compromise between allocation quality and decision latency.

The findings should not be interpreted as measured performance improvements because no empirical dataset or experimental benchmark was supplied. Rather, they represent analytically derived outcomes from the proposed architecture and its theoretical foundations.

Empirical implementation remains necessary to quantify improvements in throughput, latency, recovery time, and resource efficiency.

The model also indicates that adaptive AI coordination should be evaluated under heterogeneous priorities rather than homogeneous workloads. Weighted allocation concepts provide a theoretical basis for assigning greater importance to latency-sensitive or resilience-critical streams (Chakraborty et al., 2020). In this context, a successful system is not necessarily one that gives every stream identical computational resources, but one that allocates resources proportionally to operational requirements while preventing unacceptable service degradation.

The overall analytical result is therefore that resilience, efficiency, and fairness can be treated as interconnected objectives within an adaptive multi-agent streaming system rather than as isolated engineering functions. This conceptual direction is consistent with the broader objective of AI-based multi-agent approaches for optimized event streaming and resilient distributed processing (R. Reddy, P. Udayaraju, K. K. Goyal, S. C. R. Vudem, R. Sayana and V. Gummadi, 2026).

5. Discussion

The proposed model contributes theoretically by transferring principles from computational allocation research into a dynamic streaming context. Traditional fair-division research primarily addresses allocation among agents under specified resource constraints, whereas real-time streaming introduces temporal variability, incomplete information, and continuously changing demand. The proposed architecture therefore extends allocation from a one-time decision into a feedback-controlled process.

One important implication concerns the relationship between fairness and efficiency. A purely throughput-oriented streaming optimizer may continuously favor high-volume workloads because they produce the largest immediate performance gain. However, such behavior can starve lower-volume but latency-critical streams. Welfare-based mechanisms offer an alternative by considering aggregate system utility and distributional effects (Amanatidis et al., 2020; Caragiannis et al., 2016). Nevertheless, fairness itself introduces trade-offs. Excessive equalization may reduce the ability of the system to prioritize genuinely critical workloads. Weighted allocation is therefore more appropriate for heterogeneous streaming environments.

A second implication concerns scalability. Increasing the number of agents can improve parallel decision-making but also increases coordination complexity. Allocation research shows that seemingly simple resource-allocation problems may become computationally difficult as the system expands (Bredereck et al., 2019; de Keijzer, Bouveret, Klos, & Zhang, 2009). The proposed hierarchical architecture addresses this issue by allowing local decisions to occur close to the source of information while using higher-level coordination only when required.

A third issue is approximation. Approximate allocation can provide timely decisions, but approximation quality may deteriorate under extreme workload conditions. The challenge is therefore to determine acceptable approximation bounds for different classes of streaming workloads. Budish (2011) and Cole and Gkatzelis (2015) provide theoretical motivation for approximate allocation, but translating such guarantees directly to dynamic streaming requires further investigation.

The principal limitation of the present study is its conceptual nature. No real-time streaming dataset, hardware deployment, failure-injection experiment, or quantitative benchmark has been provided. Consequently, claims regarding performance improvement remain hypotheses rather than experimentally verified results. Additional limitations include communication overhead among agents, the possibility of unstable adaptation under rapidly changing workloads, inaccurate workload prediction, and vulnerability to corrupted monitoring information.

From a practical perspective, the proposed model is most appropriate for systems where workloads change frequently and uninterrupted processing is important. Its implementation should begin with bounded adaptation policies, explicit resource limits, and measurable recovery criteria. Future empirical research should investigate whether the additional intelligence and coordination overhead produce a statistically significant benefit over simpler scheduling mechanisms. The connection between AI-based multi-agent streaming optimization and resilience provides a promising direction for such validation (R. Reddy, P. Udayaraju, K. K. Goyal, S. C. R. Vudem, R. Sayana and V. Gummadi, 2026).

6. Conclusion

This paper proposed an adaptive AI multi-agent model for optimizing real-time data streaming and system resilience. The framework integrates monitoring, prediction, resource allocation, coordination, and fault recovery into a distributed adaptive architecture. Its theoretical foundation is derived from established research on efficient allocation, fair division, Nash welfare, approximation, and computational complexity.

The principal contribution is the conceptual

integration of allocation and resilience into a continuous feedback mechanism. Rather than treating resource management and fault recovery as independent processes, the proposed model enables system state to influence both ordinary allocation and recovery decisions. This design is particularly suitable for dynamic streaming environments in which workload intensity and resource availability can change rapidly.

The analysis indicates that welfare-oriented and approximate allocation principles can help balance system-wide performance against computational decision costs. However, the framework also exposes important trade-offs involving fairness, priority, scalability, coordination overhead, and adaptation stability. Most importantly, the present findings are analytical rather than experimentally measured.

Future research should implement the architecture in a real streaming platform and evaluate it under burst workloads, heterogeneous priorities, node failures, network degradation, and changing resource availability. Quantitative comparison with static and centralized scheduling approaches should measure throughput, latency, resource utilization, recovery time, workload loss, and coordination cost. Such experiments would determine whether the theoretical advantages of adaptive multi-agent coordination translate into measurable improvements in real-world streaming resilience and scalability.

References

1. Amanatidis, G., Aziz, H., Birmpas, G., Filos-Ratsikas, A., Li, B., Moulin, H., Voudouris, A. A., & Wu, X. (2023). Fair division of indivisible goods: Recent progress and open questions. Artif. Intell.,322, 103965.

2. Amanatidis, G., Birmpas, G., Filos-Ratsikas, A., Hollender, A., & Voudouris, A. A. (2020).Maximum Nash welfare and other stories about EFX. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI), pp. 24–30.

3. Aziz, H., Lindsay, J., Ritossa, A., & Suzuki, M. (2022). Fair allocation of two types ofchores..

4. Barman, S., Krishnamurthy, S. K., & Vaish, R. (2018a). Finding fair and efficient allocations. In Proceedings of the 19th ACM Conference on Economics and Computation (EC),pp. 557–574.

5. Barman, S., Krishnamurthy, S. K., & Vaish, R. (2018b). Greedy algorithms for maxi-mizing Nash social welfare. In Proceedings of the 17th International Conference on Autonomous Agents and Multi Agent Systems (AAMAS), p. 7–13.

6. Bliem, B., Bredereck, R., & Niedermeier, R. (2016). Complexity of efficient and envy-freeresource allocation: Few agents, resources, or utility levels. InProceedings of the 25thInternational Joint Conference on Artificial Intelligence (IJCAI), p. 102–108.

7. Brams, S., & Taylor, A. (1996).Fair Division: From Cake-Cutting to Dispute Resolution.Cambridge University Press.

8. Brˆanzei, S., & Sandomirskiy, F. (2019). Algorithms for competitive division of chores.CoRR,abs/1907.01766.

9. Bredereck, R., Kaczmarczyk, A., Knop, D., & Niedermeier, R. (2019). High-multiplicityfair allocation: Lenstra empowered by n-fold integer programming. InProceedings ofthe 2019 ACM Conference on Economics and Computation, p. 505–523.

10. Budish, E. (2011). The combinatorial assignment problem: Approximate competitive equi-librium from equal incomes.Journal of Political Economy,119(6), 1061 – 1103.

11. Caragiannis, I., Kurokawa, D., Moulin, H., Procaccia, A. D., Shah, N., & Wang, J. (2016).The unreasonable fairness of maximum Nash welfare. InProceedings of the 17th ACMConference on Economics and Computation (EC), p. 305–322.

12. Chakraborty, M., Igarashi, A., Suksompong, W., & Zick, Y. (2020). Weighted envy-freenessin indivisible item allocation. InProceedings of the 19th International Conference onAutonomous Agents and MultiAgent Systems (AAMAS), p. 231–239.

13. Cole, R., & Gkatzelis, V. (2015). Approximating the Nash social welfare with indivisibleitems. InProceedings of the Forty-Seventh Annual ACM Symposium on Theory ofComputing (STOC), p. 371–380.

14. Darmann, A., & Schauer, J. (2014). Maximizing Nash product social welfare in allocatingindivisible goods.SSRN Electronic Journal,247.

15. de Keijzer, B., Bouveret, S., Klos, T., & Zhang, Y. (2009). On the complexity of efficiencyand envy-freeness in fair division of indivisible goods with additive preferences. InRossi, F., & Tsoukias, A. (Eds.),Algorithmic Decision Theory, pp. 98–110.

16. R. Reddy, P. Udayaraju, K. K. Goyal, S. C. R. Vudem, R. Sayana and V. Gummadi, "Creating an AI-based Multi-Agent Model for Optimized Data Streaming with Improved Resiliency and Scalability for Event Streaming," 2026 6th International Conference on Image Processing and Capsule Networks (ICIPCN), Dhulikhel, Nepal, 2026, pp. 1372-1379, doi: 10.1109/ICIPCN67432.2026.11438445.

17. K. Ramamurthy, N. Bellamkonda and N. Amanmadov, "ScalePulse: Combinatorial Llm Framework for Scalability Constraint," SoutheastCon

2026, Huntsville, AL, USA, 2026, pp. 1-6, doi: 10.1109/SoutheastCon63549.2026.11476075

18. Philip, P. G. (2024). Artificial Intelligence-Driven Project Risk Prediction Models: Enhancing Decision-Making Accuracy in Large-Scale Infrastructure Projects . American Journal of Technology, 3(1), 52–69. https://doi.org/10.58425/ajt.v3i1.571

19. SinghJatav, D., Amin, M. M., Kodela, S., Nayan, V., Wannous, M., & Khalifa, G. S. (2025, November). Hybrid Reinforcement and Deep Learning Model for Payment Delay Optimization in Supply Chain Finance. In 2025 10th International Conference on Information Technology Trends (ITT) (pp. 170-175). IEEE.