International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 7, 2025
https://www.theaspd.com/ijes.php

Optimizing Network Load Distribution With Maximum Flow Algorithms In The WSCLB Framework

Paul Sheeba Ranjini[1], Tulasiram Hemamalini[2], Angalakurichi Natraj Senthilvel[3]
[1]Department of Science and Humanities (Mathematics), Sri Krishna College of Technology, Coimbatore, Tamilnadu, India
[2]Department of Mathematics, Government Arts College (Autonomous), Coimbatore, Tamilnadu, India
[3]Department of Computer Science and Engineering, Coimbatore Institute of Technology, Coimbatore, Tamilnadu, India, sheebaranjini.p@skct.edu.in

ABSTRACT
Weighed Scalable Consistent Load Balancing (WSCLB) is a framework for optimising network load distribution via the use of maximum flow methods. The goal is to enhance network efficiency by identifying the optimal distribution of network resources to prevent node overload and ensure equitable allocation of resources. The objective is to maximise throughput while minimising congestion in the network by identifying optimum channels for data delivery using maximum flow algorithms. The WSCLB framework may be enhanced with the help of these algorithms, leading to better network performance, more effective use of resources, and consistent system stability. Managing distributed network systems has never been easier than with this method, which tackles issues like scalability and load balancing head-on. In a sample of five jobs from five nodes, the first occurrence of Flow Capacity Matrix yielded the following values for 5 links: 10–40 for Node 1, 15–35 for Node 2, 10–30 for Node 3, 10–40 for Node 4, and 15–30 for Node 5. The second iteration of Optimized_Load_Distribution yields the following results using a sample of five connections from five nodes: From 55 to 94 The values for Nodes 1, 2, 3, 4, and 5 are 15, 24, 17, and 17 and 24, respectively.
Keywords: *Network Efficiency, Resource Distribution, Node Overload, Maximum Flow Algorithms WSCLB Framework*

1. INTRODUCTION

Complex distributed systems need network load distribution optimisation for efficiency and stability. Maximum Flow Algorithms, which tackle network resource flow issues, improve load distribution. These techniques improve network efficiency by balancing load across several nodes with different capabilities and demands in the WSCLB architecture. Maximum Flow Algorithms are used to optimise load distribution techniques and network resource allocation. These methods should be used in WSCLB to improve load balancing for present and future network needs. This method improves load distribution systems' scalability and flexibility, making them better resource managers in dynamic network contexts. Maximum Flow Algorithms may increase load distribution optimisation in the WSCLB framework, improving network speed and resource utilisation.
Achieving high performance and dependability in distributed systems requires optimising network load distribution. Conventional load balancing techniques may become inadequate to handle resource demands posed by increasingly sophisticated and large-scale networks. One effective method for dealing with these problems is the use of Maximum Flow Algorithms, which are algorithms that, given certain limitations, find the highest feasible network flow. Using these techniques in conjunction with the WSCLB architecture may greatly improve the efficiency of load distribution plans. The main goal is to enhance load balancing across network nodes by using Maximum Flow Algorithms inside the WSCLB framework. The goal is to maximise the overall efficiency of the network by balancing the loads on each node and making use of the algorithms' ability to optimise the flow of resources. Finding a way to distribute loads that works now and can easily expand to meet future network demands is the main objective. Improved network performance and resource utilisation are outcomes of the WSCLB framework's use of Maximum Flow Algorithms, which allow for more accurate and adaptive load balancing. By offering a more flexible and scalable solution, this method seeks to address the drawbacks of conventional load balancing techniques. Load distribution methods are anticipated to be refined by the inclusion of Maximum Flow Algorithms, leading to improved network stability and

efficiency. The capacity to optimise load distribution dynamically is going to be critical for sustaining high levels of performance and dependability as networks keep growing and become more complicated.

Work Contribution: When it comes to maintaining the stability and performance of distributed systems, optimising the distribution of network loads is essential. When it comes to today's complex and ever-changing network settings, traditional load balancing methods sometimes fall short. To tackle these issues, a big step forward has been made with the incorporation of Maximum Flow Algorithms into the WSCLB architecture. By showing how to use Maximum Flow Algorithms inside the WSCLB framework to improve load distribution techniques, this study contributes to the area. Optimising resource allocation and balancing loads among network nodes is the main emphasis of the study, which makes use of Maximum Flow Algorithms. The WSCLB framework can handle different node capacities and demands better after implementing these algorithms. The contribution is in demonstrating how to use Maximum Flow Algorithms in a real-world setting to improve network performance and efficiency via more accurate and scalable load balancing. Offering a strong approach to improving current load balancing procedures, this contribution tackles the need for enhanced solutions in network load distribution. Improving resource management and flexibility in dynamic network settings may be achieved by integrating Maximum Flow Algorithms into the WSCLB framework. Improving network speed and resilience is a direct result of optimising load distribution, which these algorithms successfully apply.

Section 2 describes how the WSCLB Framework uses the Maximum Flow Algorithms for load balancing on distributed networks. Section 3 discusses Maximum Flow Algorithms used for Load Balancing in Distributed Networks in the WSCLB Framework. The Maximum Flow Algorithms uses numerous datasets for load balancing in distributed networks in the WSCLB method, as described in Section 4. Finally, Section 5 concludes with a conclusion.

2. LITERATURE SURVEY

Evolutionary Algorithm Comparison Distribution Network Optimal PV Allocation [1]. EAs like PSO, ABC, DE, and derivatives optimise PV allocation. Performance determines effectiveness. Near-Linear Work: Polylogarithmic Depth and Parallel Approximate Maximum Flows [2]. This method allows us to derive subgraph congestion approximators from the overall graph, eliminating unnecessary recursion. A parallel flow-decomposition approach key to obtaining polylogarithmic depth, which may be of interest independently is created. In, a precise method maximises electric car flow coverage with heterogeneous chargers, nonlinear charging time, and route deviations [3]. An precise Benders decomposition-based technique uses layered cut-generation using a specialized labeling algorithm for the Benders subproblem. Two strategies for tackling this subproblem are combined to speed Benders cut creation. For bigger issues, a heuristic method based on the precise approach is suggested. Computational findings show the method solves real-world challenges. An upgraded social network search algorithm optimizes economic load dispatch in large-scale power systems globally is applied in [4]. This study offers an improved social network search (ESNS) method to meet the given goals. The SNS algorithm relies on users' discussion, imitation, originality, and disagreements. The ESNS algorithm improves the SNS technique by improving search capabilities, especially for optimal solutions. The main objective is to enhance the algorithm's global search capabilities while avoiding local optimization.

Deep Neural Networks for Power Flow Analysis in Three-Phase Unbalanced Smart Distribution Grids is described in [5]. Deep learning (DL) is used to forecast PF solutions for three-phase unbalanced power distribution grids in this work. This research proposes three deep neural networks (DNNs): Radial Basis Function Network (RBFnet), Multi-Layer Perceptron (MLP), and Convolutional Neural Network (CNN) for PF solution prediction. The suggested DNN models outperform classic iterative-based PF solvers by correctly predicting solutions by capturing nonlinear interactions in computations. To optimize power flow in DC networks, use the Whale Optimization Algorithm is described in [6]. This research proposes a DC network optimum power flow (OPF) solution technique. The master-slave optimization approach combines a whale optimization algorithm (WOA) with a numerical method using successive approximations (SA). The purpose is to minimize power losses in a dispersed generating setting, considering DC network restrictions. Maximizing distributed energy system design with multiphase optimum power flow using complementarity reformulations is described in [7]. The present work is the first to offer an optimization methodology for grid-connected DES that accounts for MOPF restrictions and appropriately represents unbalanced low-voltage distribution. An extensive review of static and dynamic distribution network reconfiguration methods is offered in [8]. Presents a thorough analysis of contemporary network reconfiguration literature. Five reconfiguration strategies are categorized: classical, heuristic, metaheuristic, hybrid, and machine learning based. The study defines, compares, and applies the categories to dynamic and static reconfiguration.

Electricity Distribution Network Design: Demand Coincidence Impact is descried in [9]. Distribution network reconfiguration problem with line-specific demand coincidence (DNRP-LSDC) is a new paradigm for building energy distribution networks. A Swiss distribution network operator applies the idea to many real-world networks. Optimizing Distribution Network Reconfiguration with a New Algorithm Hyperbolic Tangent Particle Swarm Optimization (HT-PSO) is presented in [10]. Hyperbolic tangent particle swarm optimisation (HT-PSO) is used to reduce power loss in distribution network reconfiguration (DNR). The new hyperbolic tangent function limits the rate of change values, improving this method over sigmoid function-based one. Electric car charging station placement in distribution networks is thoroughly analysed [11]. Recent energy research has made EVs more feasible. Thus, EVs have been popular and quickly adopted in many countries. The rapid spread of EVs has made issues like insufficient charging infrastructure, unequal distribution, exorbitant cost, and a lack of charging stations crucial. GAT-Based Load-Aware Network Resource Orchestration in LEO Satellite Networks [12]. Service function chain (SFC) orchestration in dynamic LEO satellite networks is examined for flexibility and efficiency. An integer nonlinear programming (INLP) problem optimises satellite service acceptance and load fairness based on service demands and network resource constraints for SFC orchestration.

Applying Power Injection-based Equations for Convex Optimal Power Flow in Bipolar DC Distribution Networks is implemented in [13]. The power flow properties of bipolar DCDN are disclosed using power injection-based equations, forming the basis of the original OPF model. The original OPF model is transformed into a convex model using SOCP, including variable substitution, relaxation, McCormick relaxation, and Taylor expansion. Reinforcement learning on graphs manages active distribution network outages in real time [14]. Distribution network resilience is increased via graph reinforcement learning outage management. This graph learning method uniquely handles network design, switching management, and complex state variable interdependencies across nodes and edges. Deep learning and optimisation are used in this hybrid cloud load balancing and host consumption prediction technique [15]. A hybrid model (DPSO-GA) integrating deep learning, Particle Swarm Intelligence, and Genetic Algorithm for dynamic cloud workload provisioning is presented in this paper. The proposed model is two-stage. For prediction, a hybrid PSO-GA technique optimises Hyperparameters leveraging the benefits of both approaches. CNN uses LSTM. PSO-GA is used to train CNN-LSTM before forecasting resource utilisation. A one-dimensional CNN and LSTM forecast cloud resource utilisation over time in the proposed architecture. Off-policy Content Marketplace Learning It implies ad-load balancing in [16]. A method using recorded bandit feedback for off-policy learning and assessment is provided. Beginning with a compelling look at the ad-load balancing challenge, the study highlights the contradiction between user delight and ad income. User heterogeneity and session position are highlighted.

Detailed literature overview on soft computing methods for cloud load balancing networks is proposed in [17]. A thorough literature review of computational paradigms-based cloud load balancing strategies is conducted. The project aims to assess the reliability of various strategies in balancing load in dynamic cloud environments. Cloud Computing Resource Allocation Optimization using Machine Learning is implied in [18]. Large-scale distributed computing relies on resource allocation as computer networks solve complicated optimization issues. In this example, resource allocation aims to maximize computer throughput. Distributed computing has two primary types: grid and cloud. Grid computing links numerous geographically scattered clusters for public use. A systematic review of cloud architectural approaches to optimise TCO and resource utilisation for high service availability and rapid elasticity [19]. This systematic research synthesises cloud architectural approaches to optimise TCO, resource use, service availability, and rapid adaptability. The study uses a decade-long systematic evaluation of peer-reviewed journals, conference papers, and relevant industry reports. VMMISD, a virtual machine migration load balancing model, uses fused metaheuristics, iterative security measures, and deep learning optimisations [20]. The algorithms were chosen for their ability to emulate natural processes to tackle complex optimisation problems. Additionally, our contextual side chaining solution leverages deep reinforcement learning to boost security. Contextual sidechaining addresses increasing security problems and connects important events, preventing further assaults.

Distributed System with Auto Promote and Web Services installation [21]. This research uses database replication, specifically multi-master, to improve availability and reduce failure risk. This research replicates PostgreSQL databases in distributed transaction distribution using auto-promote master and web services. This research examines real-world college site selection transactions that need speed, scalability, and data. The early user model design for cognitive load-based adaptive interface development in learning management systems is [22]. LMS stagnates and can't adapt to learners' abilities without cognitive process understanding. Research is required to develop the cognitive load model that best portrays LMS users as adaptive interface triggers. We interviewed four cognitive experts. Implementing a Distributed Cluster System Performance Optimisation Strategy Case Study in [23]. To address this issue, we must design a high-performance Distributed Cluster System (DCS) with the appropriate architecture. Each cluster has its own VLAN and function. This research combines case study and system development approaches, with post-implementation assessment. All features of built-in technology are observed. This study highlights performance concerns caused by monoliths and a cluttered architecture. Event Based System (EBS) helps DCS handle peak processing workloads. Web Application Penetration Testing on Udayana University's OASE E-learning Platform using Information System Security Assessment Framework (ISSAF) and Open-Source Security Testing Methodology Manual (OSSTMM) is proposed in [24]. Web-based technologies are prone to attacks, thus effective security is essential. Udayana University uses OASE, a web-based instructional program. Because OASE is web-based, it needs extensive security verification. Penetration testing evaluates OASE security. This testing may be done using ISSAF and OSSTMM. Review of Hybrid and Federated Cloud Architectures for Distributed Multi-Cloud Computing is presented in [25]. The research examines the pros and cons of managing several cloud providers. The overview begins with multi-cloud basics and emphasises the importance of flexibility, scalability, and resilience in modern computing.

3. METHODS AND MATERIALS

3.1. Using Maximum Flow Algorithms in WSCLB Frameworks to Address Load Distribution Issues

Uneven load distribution in distributed network systems may overwhelm servers, reduce performance, and waste resources. Strong load-balancing solutions are needed to handle network traffic and overcome these issues. A solution is to use maximum flow methods in Weighted Server and Client Load Balancing (WSCLB). These algorithms find the best data flow channels between network nodes to balance traffic on each server. Maximum flow algorithms may find bottlenecks and redistribute workloads by portraying the network as a flow network with linked nodes, decreasing server overload. These methods enable real-time, dynamic load distribution modifications in WSCLB frameworks, improving reaction times and system dependability. Mathematics is used to effectively solve load-balancing problems, optimising server resource use and network service quality. Maximum flow methods solve dispersed network load distribution problems scalable and effectively by optimising flow.

The starting condition of a distributed network with jobs given to nodes is shown in Figure 1. Nodes A, B, and C send their jobs to the control layer, which monitors network task allocation. The graphic shows how jobs are originally divided, causing imbalances where some nodes are overloaded, and others underutilised. This situation allows the control layer to optimise work distribution. This section focusses on node load and how it may be inefficient. This is the crucial first step before using the Maximum Flow algorithm in WSCLB to balance and optimise network load distribution.

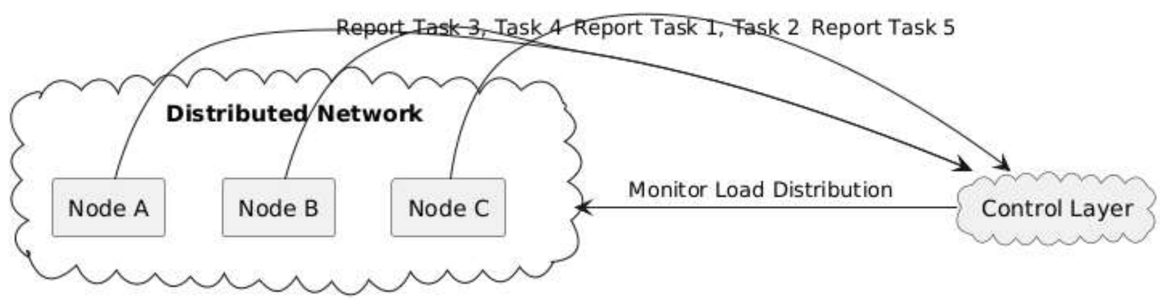

Figure 1. Initial Network Load and Task Distribution

3.2. How Maximum Flow Algorithms Optimise Network Load Distribution in WSCLB Frameworks

Distributed network systems need load distribution to avoid server overload, assure performance, and maximise resource use. Maximum flow methods help Weighted Server and Client Load Balancing (WSCLB) systems achieve these goals. These algorithms analyse the network as a flow network to maximise data flow between nodes without congestion. Maximum flow methods dynamically balance server loads by finding the most efficient data transmission pathways. Dynamic adjustment decreases latency, boosts throughput, and increases system responsiveness. WSCLB frameworks with maximum flow algorithms optimise resource utilisation, reduce network bottlenecks, and increase service dependability. These algorithms scale complicated network topologies and traffic needs. Their mathematical accuracy helps handle enormous data sets and balance load distribution even under strong demand. Maximum flow algorithms in WSCLB frameworks improve network performance and efficiency, making them useful in current distributed systems. Figure 2 shows how the WSCLB framework uses the Maximum Flow method to optimise workload allocation across network nodes. The control layer analyses node load using the Maximum Flow technique after spotting imbalances in the initial job allocation. The program then adjusts tasks to minimise overloads and assure work distribution. For instance, overloaded nodes' duties are moved to lighter ones. The technique optimises job flow between nodes to reduce communication costs and boost network efficiency. This figure shows how the Maximum Flow algorithm analyses and resolves load imbalances to improve distributed network job allocation.

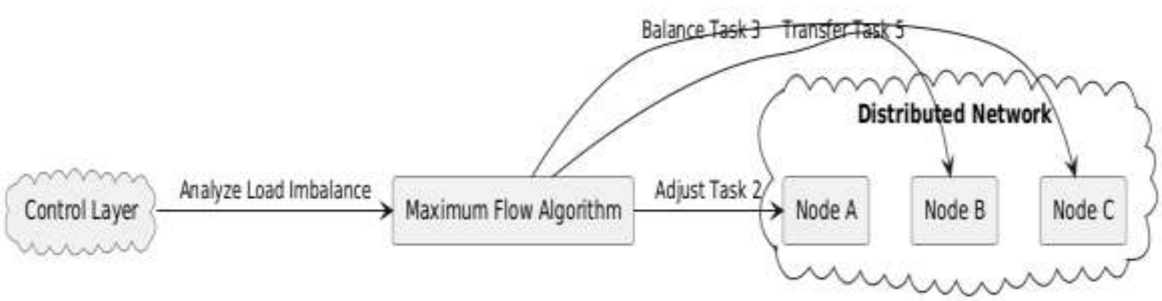

Figure 2. Application of Maximum Flow Algorithm for Load Optimization

3.3. Maximum Flow Algorithms' Effects and Limitations on WSCLB Network Load Distribution Optimisation

Maximum flow algorithms improve network load distribution by balancing dispersed system traffic. These techniques help reduce server overload, latency, and network performance in Weighted Server and Client Load Balancing (WSCLB) frameworks. Maximum flow algorithms dynamically allocate resources by determining the best data pathways, improving utilisation and scalability. Maintaining service quality in dispersed networks requires faster reaction times, better throughput, and more dependable service delivery. However, these algorithms have drawbacks. Solving maximum flow issues in real time is computationally difficult, especially in large networks with quickly changing circumstances. Traditional maximum flow algorithms' static nature makes them less efficient in contexts that demand continual modifications due to network traffic fluctuations. Network topology data must be precise and current to avoid inefficient load distribution. The Maximum Flow algorithm in the WSCLB framework results in the network state shown in Figure 3. Tasks are uniformly spread among all nodes to avoid overloading or underutilising anyone. The control layer ensures optimised task allocation after implementing the algorithm's suggestions. A, B, and C have balanced workloads with the right amount of jobs. Balance minimises network latency, improves performance, and optimises resource use. The graphic shows a highly efficient network after optimisation. A stable and successful distribution network system requires optimised load distribution.

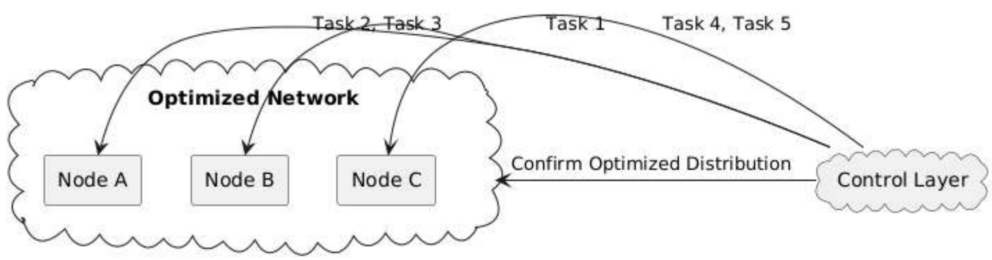

Figure 3. Optimized Load Distribution

3.4. Methods and Techniques of Maximising Flow Algorithms for Network Load Distribution

Distributed systems need appropriate network load distribution for performance and dependability. Maximum flow algorithms find the optimal data transmission pathways, guaranteeing that no server is underutilised or overburdened. Various approaches and strategies apply maximum flow algorithms in network load distribution. The Ford-Fulkerson method, which uses augmenting paths to find a flow network's maximum flow, is fundamental. Increase traffic until no more augmenting channels are available to balance network pressure. Another prominent method for maximum flow solutions is the Ford-Fulkerson Edmonds-Karp algorithm, which uses breadth-first search to find shortest augmenting paths. Another method, Push-Relabel, pushes excess flow throughout the network and relabels nodes to allow flow. This approach works well for complicated topologies and networks with different capabilities. Advanced methods like capacity scaling and dynamic tree data structures improve these algorithms' performance, making network load distribution systems quicker and more scalable. These approaches provide a complete network traffic and resource allocation toolbox. Figure 4 shows a simplified data flow diagram (DFD) of the Maximum Flow algorithm's WSCLB network load distribution optimisation process. Starting with the user giving tasks to the dispersed network. Each network node communicates its job load to the control layer, which analyses load distribution. Maximum Flow is used in the control layer to find imbalances and optimise work allocation. The control layer changes network node task allocation after optimisation. Finally, nodes complete duties and notify the user. This graphic shows the user, distributed network, and control layer interactions for optimised load distribution

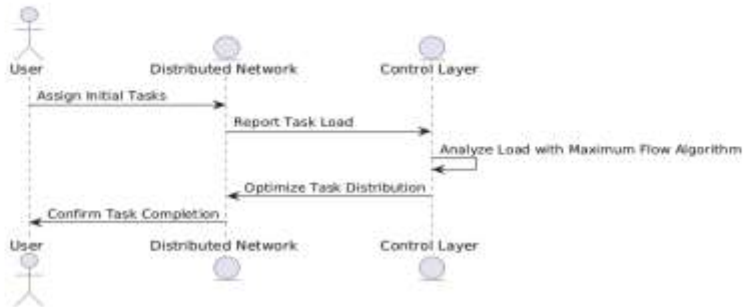

Figure 4. Data Flow Diagram for Optimizing Network Load Distribution Using Maximum Flow Algorithm in the WSCLB Framework

3.5. Flow Conservation Equation

$$\sum_{\mathbf{v} \in \mathbf{V}} \mathbf{f}(\mathbf{u}, \mathbf{v}) = \mathbf{0}, \ \forall \ (\mathbf{u} \ \in \mathbf{V}) - \{\mathbf{s}, \mathbf{t}\} \qquad (1)$$

This equation 1 states that for every node u (except the source s and the sink t), the total flow into the node equals the total flow out. It ensures that no flow is lost within the network, maintaining consistency in data transmission.

3.6. Capacity Constraint Equation

$$0 \ \leq f(u, v) \leq c(u, v), \forall \ (u, v) \ \in E \qquad (2)$$

4. RESULTS AND DISCUSSION

Types of Maximum Flow Algorithms in Network Load Distribution

4.1. Ford-Fulkerson Algorithm

The Ford-Fulkerson algorithm is a basic approach for determining maximum flow in a flow network. Augmenting routes searches for source-to-sink pathways that can handle more flow. The technique efficiently maximises network flow by augmenting flow along these pathways until no more augmenting paths are discovered. Ford-Fulkerson is flexible but can have high time complexity, especially with irrational flow values.Figure 5 shows how the Maximum Flow method in WSCLB (Weighted Sum Cost Load Balancing) optimises network load distribution. Nodes A, B, and C handle particular duties in the distributed network. The control layer regularly monitors and analyses task load data from these nodes. Load distribution and imbalances are assessed using the WSCLB technique. The Maximum Flow method then creates an ideal task reallocation plan to equally distribute tasks across network nodes. Finally, the control layer adjusts task allocations based on the optimised plan for a balanced, efficient network. This image shows the key phases and interactions for distributed network load distribution optimisation.

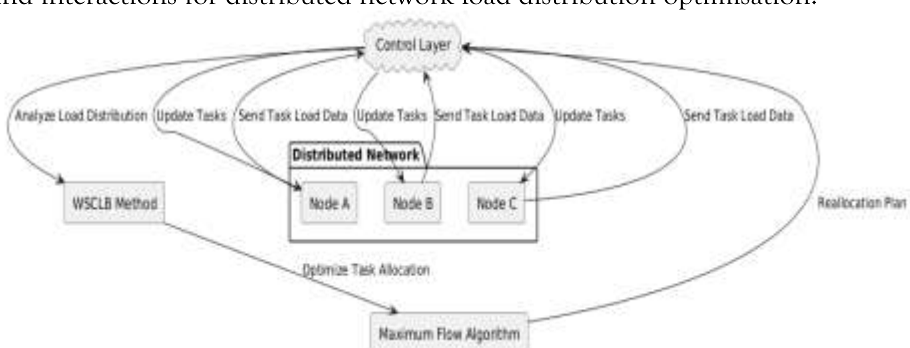

Figure 5. Overview of Network Load Optimization Using Maximum Flow Algorithm in WSCLB Framework

4.2. Edmonds-Karp Algorithm

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 7, 2025
https://www.theaspd.com/ijes.php

An implementation of the Ford-Fulkerson method, the Edmonds-Karp algorithm utilizes Breadth-First Search (BFS) to find the shortest augmenting paths in terms of the number of edges. This optimization leads to a time complexity of$O\ (VE^2)$, where V is the number of vertices and E is the number of edges. Edmonds-Karp is known for its simplicity and effectiveness, particularly in networks with relatively small numbers of vertices and edges.Table 1 shows a maximum flow algorithm network load distribution scenario. The network has five nodes (A, B, C, D, and E) with Mbps flow capacity and connections. The initial flow column indicates the beginning flow, while the additional flow column shows the maximum flow algorithm-calculated flow. Total flow is the sum of original and extra flows, showing the method optimises network traffic. The connection from node A to node B starts at 20 Mbps and adds 10 Mbps, totalling 30 Mbps. This optimises network performance by using each network route effectively without exceeding its capacity.

Table 1. Sample Network Load Distribution Using Maximum Flow Algorithms

Node Pair	Capacity (Mbps)	Initial Flow (Mbps)	Additional Flow (Mbps)	Total Flow (Mbps)
A → B	50	20	10	30
A → C	40	15	15	30
B → D	60	25	20	45
C → D	70	30	25	55
D → E	80	40	30	70

4.3. Push-Relabel Algorithm

Unlike augmenting path algorithms, the Push-Relabel algorithm focuses on maintaining a preflow, which can exceed the capacities temporarily. It uses local operations to push flow from higher to lower nodes and relabels nodes to create new flow paths. This algorithm is well-suited for networks with dense graphs and achieves good performance with a time complexity of$O\ (VE^2)$.Figure 6 shows the flow capacity between dispersed network nodes and connections. Each column in the matrix reflects a node's maximum flow capacity across a connection (in arbitrary units). Maximum flow algorithms optimise network load distribution in WSCLB (Weighted Scalable Consistent Load Balancing) using this data. The program analyses flow capacity to find the best network load distribution pathways, guaranteeing no connection or node is overburdened. This matrix emphasises the need to know the network's capacity limitations to preserve efficiency and eliminate bottlenecks for equitable load distribution.

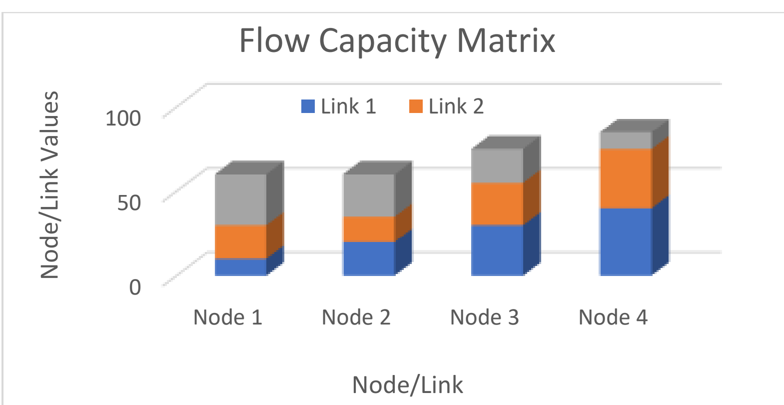

Figure 6. Flow Capacity Matrix for Network Load Distribution

Table 2 lists the key responsibilities, advantages, functions, and pros and drawbacks of maximum flow algorithms for network load distribution. Data integrity is maintained via flow conservation, while capacity limitations prevent network congestion by conforming to preset limits. By adding data flow pathways, augmenting paths optimise resource consumption. Adjusting to changing network circumstances in real time improves responsiveness and flexibility. These approaches need powerful processing and real-time monitoring because to their computational complexity. These factors show the pros and cons of maximum flow methods in network load allocation.

Table 2. Aspects of Network Load Distribution with Maximum Flow Algorithms

Aspect	Role	Benefit	Function	Pros	Cons
Flow Conservation	Maintains flow consistency in the network	Prevents data loss	Ensures incoming flow equals outgoing flow	Preserves network stability	May require complex calculations
Capacity Constraints	Limits flow based on capacity	Avoids overload	Restricts flow to not exceed link capacity	Protects network from congestion	Requires accurate capacity data
Augmenting Paths	Finds new paths for additional flow	Increases network utilization	Searches for paths to enhance flow	Optimizes resource use	Can be time-consuming
Real-Time Adjustment	Dynamically changes flow distribution	Adapts to network changes	Modifies flow allocation as network changes	Improves response to fluctuating demands	Requires real-time monitoring
Computational Complexity	Measures processing effort	Ensures algorithm efficiency	Evaluates complexity for scalability	Provides scalability insights	May limit algorithm applicability

4.4. Dinic's Algorithm:

Dinic's algorithm introduces the concept of level graphs to optimize the search for augmenting paths. It breaks the network into different levels using BFS, and then finds all possible augmenting paths at the current level before moving to the next. This method significantly reduces the number of paths that need to be explored and has a time complexity of$O\ (VE^2)$. Dinic's algorithm is efficient for networks with large numbers of vertices and edges.Figure 7 shows the optimised load distribution across nodes after using WSCLB maximum flow methods. The matrix values indicate each node's optimised load (in arbitrary units), guaranteeing network balance. The ideal load pathways were calculated using the maximum flow method and the flow capacity from the preceding figure. The idea is to optimally distribute network load to reduce congestion and increase throughput. This optimised matrix shows how maximum flow methods may improve network performance by ensuring each node handles enough load, boosting distributed system stability and scalability.

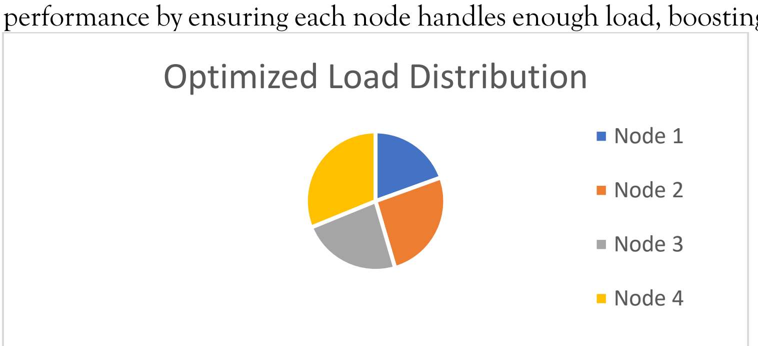

Figure 7. Optimized Load Distribution Matrix Using Maximum Flow

4.5. Capacity Scaling Algorithm:

The capacity scaling algorithm is a variation of the Ford-Fulkerson method that improves efficiency by focusing on paths with large capacities first. It scales down the capacities incrementally, optimizing the flow in stages. This approach can lead to faster convergence in networks where capacity varies widely, making it a practical choice for certain types of network load distribution problems.

5. CONCLUSION

Optimising network load distribution using maximum flow algorithms in the WSCLB framework is difficult due to the computational complexity of large-scale flow issues and the dynamic nature of network circumstances that may impact method performance. Implementing these techniques greatly improves network efficiency, load distribution, and system dependability. Scalability concerns and the need for reliable, real-time data to optimise performance are drawbacks. Future research may refine algorithms to handle bigger and more complicated networks, use machine learning to forecast and react to dynamic load circumstances, and integrate the framework with new technologies for scalability and flexibility. Addressing these difficulties and investigating future improvements will improve distributed system network load distribution tactics. According to the first occurrence of Flow_Capacity_Matrix in a sample of five activities from five nodes, the values for links 1, 2, 3, 4, and 5 are 10-40, 15-35, 10-30, 10-40, and 15-30, respectively. The second Optimized_Load_Distribution instance gives the following results using 5 connections from 5 nodes: 55-94 Nodes 1–5 have 15-25, 17-24, 16-22, 15-24, and 17-24, respectively.

REFERENCES

[1] W. Bai, W. Zhang, R. Allmendinger, I. Enyekwe, and K. Y. Lee, "A Comparative Study of Optimal PV Allocation in a Distribution Network Using Evolutionary Algorithms," *Energies*, vol. 17, no. 2, pp. 1-19, 2024. https://doi.org/10.3390/en17020511.

[2] A. Agarwal, S. Khanna, H. Li, P. Patil, C. Wang, N. White, and P. Zhong, "Parallel approximate maximum flows in near-linear work and polylogarithmic depth," *In Proc. of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 3997-4061, 2024. https://doi.org/10.1137/1.9781611977912.140.

[3] H. Park, and C. Lee, "An exact algorithm for maximum electric vehicle flow coverage problem with heterogeneous chargers, nonlinear charging time and route deviations," *European J. of Operational Res.*, vol. 315, no. 3, pp. 926-951, 2024. https://doi.org/10.1016/j.ejor.2023.12.019

[4] M. H. Hassan, S. Kamel, F. Jurado, and U. Desideri, "Global optimization of economic load dispatch in large scale power systems using an enhanced social network search algorithm," *Int. J. of Elect. Power & Energy Syst.*, vol. 156, pp. 2024. https://doi.org/10.1016/j.ijepes.2023.109719

[5] D. Tiwari, M. J. Zideh, V. Talreja, V. Verma, S. K. Solanki, and J. Solanki, "Power Flow Analysis Using Deep Neural Networks in Three-Phase Unbalanced Smart Distribution Grids," *IEEE Access*, vol. 12, pp. 29959-29970, 2024. https:// r 10.1109/ACCESS.2024.3369068

[6] S. C. Hernandez, L. F. Noreña, J. J. Montano, O. D. Montoya, and W. G. Gonzalez, "Optimal Power Flow in DC Networks Using the Whale Optimization Algorithm," *Statistics, Optimization & Information Comput.*, vol. 12, no. 5, pp. 1476-1491, 2024. https://doi.org/10.19139/soic-2310-5070-1253

[7] I. De Mel, O. V. Klymenko, and M. Short, "Complementarity reformulations for the optimal design of distributed energy systems with multiphase optimal power flow," *Int. J. of Elect. Power & Energy Syst.*, vol. 155, no. 2, pp. 1-12, 2024. https://doi.org/10.1016/j.ijepes.2023.109610

[8] M. R. Behbahani, A. Jalilian, A. Bahmanyar, and D. Ernst, "Comprehensive Review on Static and Dynamic Distribution Network Reconfiguration Methodologies," *IEEE Access*, vol. 12, pp. 9510-9525, 2024, doi: 10.1109/ACCESS.2024.3350207.

[9] G. Gust, A. Schlüter, S. Feuerriegel, I. Úbeda, J. T. Lee, and D. Neumann, "Designing electricity distribution networks: The impact of demand coincidence," *European J. of Operational Res.*, vol. 315, no. 1, pp. 271-288, 2024. https://doi.org/10.1016/j.ejor.2023.11.029

[10] D. W. Puma, Y. P. Molina, B. A. Atoccsa, J. E. Luyo, and Z. Ñaupari, "Distribution Network Reconfiguration Optimization Using a New Algorithm Hyperbolic Tangent Particle Swarm Optimization (HT-PSO)," *Energies*, vol. 17, no. 15, pp. 1-13, 2024. **https://doi.org/10.3390/en17153798**

[11] T. Yuvaraj, K. R. Devabalaji, J. A. Kumar, S. B. Thanikanti, and N. Nwulu, "A comprehensive review and analysis of the allocation of electric vehicle charging stations in distribution networks," *IEEE Access*, vol. 12, pp. 5404-5461, 2024.

[12] J. He, N. Cheng, Z. Yin, H. Zhou, C. Zhoua, K. Aldubaikhy, A. Alqasir, and X. S. Shen, "Load-aware network resource orchestration in LEO satellite network: A GAT-based approach," *IEEE Internet of Things J.*, vol. 11, no. 9, pp. 15969- 15984, 2024.

[13] Y. Zhou, Q. Wang, T. Huang, J. Liao, Y. Chi, N. Zhou, X. Xu, and X. Zhang, "Convex optimal power flow based on power injection-based equations and its application in bipolar DC distribution network," *Electric Power Systems Res.*, vol. 230, pp. 1-10, 2024. https://doi.org/10.1016/j.epsr.2024.110271

[14] R. A. Jacob, S. Paul, S. Chowdhury, Y. R. Gel, J. Zhang, "Real-time outage management in active distribution networks using reinforcement learning over graphsm," *Nature Commun.*, vol. 15, no. 1, pp. 1-17, 2024. https://doi.org/10.1038/s41467-024-49207-y.
[15] S. Simaiya, U. K. Lilhore, Y. K. Sharma, K.B. Rao, V. V. Maheswara Rao, A. Baliyan, A. Bijalwan, and R. Alroobaea, "A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques," *Scientific Rep.*, vol. 14, no. 1, pp. 1-18, 2024. https://doi.org/10.1038/s41598-024-51466-0
[16] H. Sagtani, M. G. Jhawar, R. Mehrotra, and O. Jeunen, "Ad-load Balancing via Off-policy Learning in a Content Marketplace," *InProc. of the 17th ACM Int. Conf. on Web Search and Data Mining*, pp. 586-595, 2024. https://doi.org/10.1145/3616855.3635846
[17] S. Negi, D. P. Singh, and M. M. Rauthan, "A systematic literature review on soft computing techniques in cloud load balancing network," *Int. J. of Syst. Assurance Eng. and Management*, vol. 15, no. 3, pp. 800-838, 2024. https://doi.org/10.1007/s13198-023-02217-3
[18] J. Jeyaraman, S. V. Bayani, and J. N. Malaiyappan, "Optimizing Resource Allocation in Cloud Computing Using Machine Learning," *European J. of Tech.*, vol. 8, no. 3, pp. 12-22, 2024. https://doi.org/10.47672/ejt.2007
[19] L. M. Gutta, "A Systematic Review of Cloud Architectural Approaches for Optimizing Total Cost of Ownership and Resource Utilization While Enabling High Service Availability and Rapid Elasticity," *Int. J. of Statistical Comput.and Simulation*, vol. 16, no. 1, pp. 1-20, 2024. https://journals.threws.com/index.php/IJSCS/article/view/238
[20] M. G. Brahmam, and R. V. Anand, "VMMISD: An Efficient Load Balancing Model for Virtual Machine Migrations via Fused Metaheuristics With Iterative Security Measures and Deep Learning Optimizations," *IEEE Access*, vol. 12, pp. 39351-39374, 2024. https:// 10.1109/ACCESS.2024.3373465
[21] W. Uriawa, S. Nurrobianti, T. M. Saif, R. I. Widodo, and Y. R. Asgari, "Implementing Distributed System using Auto Promote and Web Services," *Preprints*, vol. 2024. pp. 1-21, 2024. https://doi.org/10.20944/preprints202406.1995.v1
[22] M. Suryani, D. I. Sensuse, H. B. Santoso, R. F. Aji, S. Hadi, R. R. Suryono, and Kautsarina, "An initial user model design for adaptive interface development in learning management system based on cognitive load," *Cognition, Tech. & Work*, pp. 1-20, 2024. https://doi.org/10.1007/s10111-024-00772-8
[23] T. R. Anggara, "A Case Study of Implementation Strategy for Performance Optimization in Distributed Cluster System," *J. of Computer Science and Tech. Studies*, vol. 6, no. 1, pp. 233-240, 2024. https:// 10.32996/jcsts
[24] I. G. Wijaya, G. M. Sasmita, and I. P. Pratama, "Web Application Penetration Testing on Udayana University's OASE E-learning Platform Using Information System Security Assessment Framework (ISSAF) and Open-Source Security Testing Methodology Manual (OSSTMM)," *I.J. Information Tech. and Comput. Science*, vol. 2, pp. 45-56, 2024. https:// 10.5815/ijitcs.2024.02.04
[25] K. J. Merseedi, and S. R. Zeebaree, "The cloud architectures for distributed multi-cloud computing: a review of hybrid and federated cloud environment," *Indonesian J. of Comput. Science*, vol. 13, no. 2, pp. 1644-1673, 2024. https://doi.org/10.33022/ijcs.v13i2.3811