Jurnal Teknik Informatika C.I.T Medicom, 16 (2) (2024) 70-81

Published by: Institute of Computer Science (IOCS)

Jurnal Teknik Informatika C.I.T Medicom

Journal homepage: www.medikom.iocspublisher.org

Graph-based Exploration for Mining and Optimization of Yields (GEMOY Method)

Hengki Tamando Sihotang[1], Fristi Riandari[2], and Jonhariono Sihotang [3]

[1),3)] Sistem Informasi, Universitas Putra Abadi Langkat, Stabat, Indonesia

[2)] Manajemen Informatika, Politeknik Negeri Medan, Medan, Indonesia

Article Info

Article history

Received : Mar 10, 2024
Revised : Apr 19, 2024
Accepted : May 28, 2024

Keywords:

Graph-based Optimization;
GEMOY Method;
Industrial Efficiency;
Resource Allocation;
Yield Management.

Abstract

This research explores the application of graph-based optimization techniques to enhance yield management and minimize transportation costs in industrial operations, particularly focusing on mining. By representing mining sites and processing plants as nodes and transportation routes as edges in a graph, we formulated an optimization problem aimed at maximizing yields while minimizing associated costs. Utilizing linear programming, we demonstrated significant cost savings, reducing transportation costs from 2100 units to 1700 units through optimized flow distribution. The study integrates elements of graph theory, optimization algorithms, and machine learning, providing a robust framework for efficient resource allocation and operational planning. The numerical example underscores the practical applicability of these techniques, paving the way for further research and refinement to accommodate additional constraints and dynamic changes in resource availability. This research highlights the potential of graph-based methods to achieve substantial economic and operational improvements across various industrial contexts.

Corresponding Author:

Hengki Tamando Sihotang,
Program Studi Sistem Informasi,
Universitas Putra Abadi Langkat,
Jl. Letjen R. Soeprapto No.10, Kec. Stabat, Kabupaten Langkat, Sumatera Utara 20814, Indonesia,
Email: hengkitamando26@gmail.com

This is an open access article under the CC BY-NC license.

1. Introduction

In today's competitive landscape, industries such as mining, agriculture, manufacturing, and data analytics face significant pressure to optimize yields and enhance operational efficiency[1], [2]. Traditional methods often fall short in addressing the complexities and interdependencies inherent in these systems[3]. Graph-based exploration offers a promising alternative, providing a structured approach to model, analyze, and optimize intricate networks of relationships and dependencies[4], [5]. This research aims to develop and validate comprehensive graph-based methodologies to optimize yields across various applications, ultimately enhancing productivity and reducing costs[6].

The interconnected nature of modern industrial systems necessitates advanced techniques to capture and optimize their complexity[7]. Graph theory, with its ability to represent entities (nodes) and their relationships (edges), is well-suited for this purpose[8]–[10]. In mining, this could mean optimizing the extraction and transportation of resources; in agriculture, it might involve efficient water distribution or pest management; in manufacturing, streamlining production processes; and in data analytics, identifying patterns and making data-driven decisions[11]–[13]. Despite its potential, the application of graph-based methods to yield optimization remains underexplored, creating a need for focused research in this area[14].

The primary challenges in optimizing yields using traditional methods include handling complex, dynamic interdependencies, and scaling solutions to large datasets[15]–[19]. Current approaches often lack the flexibility to adapt to changing conditions and the capacity to integrate diverse data sources effectively[20]. There is a critical need for methodologies that can model these complexities accurately and provide actionable insights for optimization[21]. This research seeks to address these gaps by developing graph-based models and algorithms tailored to the specific needs of various industries[22], [23].

Previous research has demonstrated the efficacy of graph-based methods in specific contexts[24], [25]. For example, graph algorithms like Dijkstra's for shortest paths have been successfully applied in logistics and transportation optimization[26]. Max flow algorithms have been used in network capacity optimization, and graph neural networks (GNNs) have shown promise in predictive modeling and recommendation systems[27]–[29]. However, a comprehensive framework that integrates these approaches for yield optimization across multiple industries is still lacking. This research builds on these foundations, aiming to generalize and expand their application.

Graph theory provides the mathematical foundation for this research, offering tools and algorithms to model complex systems[30]–[32]. Key concepts include nodes and edges, weighted and unweighted graphs, directed and undirected graphs, and various graph traversal and optimization algorithms. By leveraging these principles, the research will develop tailored solutions for specific optimization problems. Additionally, integrating optimization techniques such as linear programming, genetic algorithms, and machine learning will enhance the ability to address non-linear and dynamic challenges effectively.

The main objective of this research is to develop and validate comprehensive graph-based methodologies to model, analyze, and optimize complex systems across various industries, thereby improving yields, enhancing productivity, and reducing operational costs. This includes constructing accurate graph representations, designing efficient algorithms, integrating optimization techniques, and applying these methods to real-world scenarios to ensure their effectiveness and practicality.

The anticipated benefits of this research include significant enhancements in productivity and operational efficiency across various industries, such as mining, agriculture, and manufacturing, through the optimization of yields. By identifying and eliminating inefficiencies, the research is expected to lead to substantial reductions in operational costs. Additionally, the development of comprehensive graph-based methodologies will provide actionable insights, facilitating more informed and strategic decision-making. The research will offer a robust, scalable, and adaptable framework for addressing complex optimization problems, applicable across different sectors and large datasets. This framework will also contribute to more sustainable and environmentally friendly operations by optimizing resource use and process flows. Furthermore, the research will advance the field of graph theory and its practical applications, contributing valuable knowledge and methodologies to both academic and professional communities. Industry practitioners will benefit from documented best practices and guidelines for implementing these optimization techniques, ultimately enhancing collaboration between academia and industry and promoting the adoption of innovative optimization strategies in industrial practices.

2. Research Methods

This research will employ a structured and systematic approach to develop and validate graph-based methodologies for optimizing yields across various industries[33]. The methodology consists of several key phases[34]:

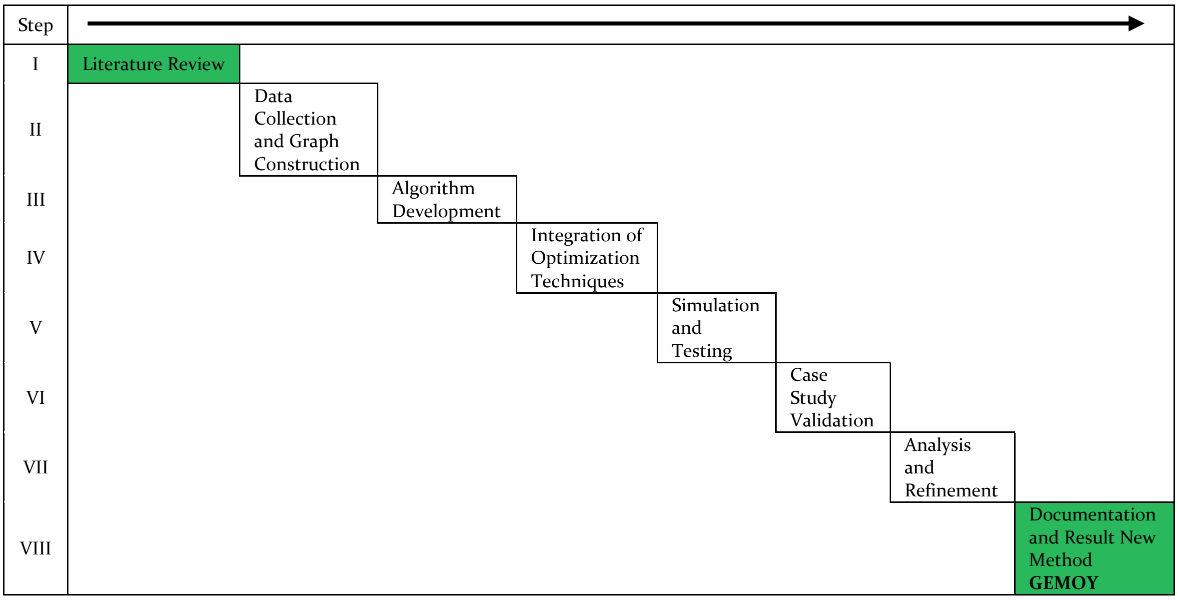

Figure 1. GEMOY Method development flow[33]

1) Literature Review:
   - Conduct a comprehensive review of existing literature on graph theory, optimization techniques, and their applications in mining, agriculture, manufacturing, and data analytics.
   - Identify gaps in current research and establish the theoretical foundation for the proposed methodologies.
2) Data Collection and Graph Construction:
   - Gather relevant data from industry partners and public sources, including information on resource allocation, production processes, logistical networks, and other operational metrics.
   - Construct graph representations of these systems, with nodes representing entities (e.g., mining sites, production units) and edges representing relationships or interactions (e.g., transportation routes, dependencies).
3) Algorithm Development:
   - Design and implement efficient graph algorithms tailored to the specific optimization problems identified in each industry.
   - Focus on algorithms for shortest paths, max flow/min cut, graph traversal (DFS, BFS), and other relevant techniques.
   - Integrate these algorithms with optimization methodologies such as linear programming, genetic algorithms, and machine learning models.
4) Integration of Optimization Techniques:
   - Combine graph-based models with optimization techniques to address non-linear and dynamic challenges.
   - Utilize machine learning models, including graph neural networks (GNNs), to enhance predictive capabilities and decision-making processes.
   - Implement reinforcement learning for dynamic process optimization by learning optimal strategies through trial and error.
5) Simulation and Testing:
   - Develop simulation environments to test the developed algorithms and models under various scenarios.
   - Conduct extensive testing to evaluate the performance, scalability, and robustness of the methodologies.

- Use synthetic data and real-world data from case studies to validate the models and algorithms.

6) Case Study Validation:
   - Apply the developed models and algorithms to real-world scenarios in different industries, including mining, agriculture, and manufacturing.
   - Collaborate with industry partners to implement and monitor the proposed solutions.
   - Collect and analyze performance data to validate the effectiveness of the methodologies and refine them as necessary.
7) Analysis and Refinement:
   - Analyze the results from simulations and case studies to identify strengths and weaknesses of the proposed methodologies.
   - Refine the graph models, algorithms, and optimization techniques based on the analysis.
   - Document best practices, guidelines, and lessons learned to inform future research and practical applications.
8) Documentation and Dissemination:
   - Prepare detailed documentation of the research process, methodologies, results, and findings.
   - Publish research papers in academic journals and present findings at conferences to disseminate knowledge.
   - Develop practical guidelines and toolkits for industry practitioners to facilitate the adoption of graph-based optimization techniques.

Basic Formulation

Graph-based exploration leverages graph theory[31], a branch of mathematics focused on the study of graphs, which are structures used to model pairwise relations between objects. This theory provides powerful tools for representing and analyzing complex systems across various domains such as mining, agriculture, manufacturing, and data analytics. The fundamental elements of graph theory—nodes (or vertices) and edges (or links)—allow for the effective modeling of entities and their interactions within a system.

Graph Representation

A graph G is defined as an ordered pair $G = (V, E)$, where:

- V is a set of vertices (or nodes), $V = \{v_1, v_2, \dots\ v_n\}$
- E is a set of edges (or links), $E = \{e_1, e_2, \dots\ e_m\}$, where each edge e is an unordered pair of vertices (for undirected graphs) or an ordered pair of vertices (for directed graphs).

Types of Graphs

- Undirected Graph: The edges have no direction. If $(v_i, v_j) \in E$, then $(v_j, v_i) \in E$.
- Directed Graph (DiGraph): The edges have a direction. If $(v_i, v_j) \in E$, then $(v_j, v_i) \notin E$ necessarily.
- Weighted Graph: Each edge e has an associated weight $w(e)$, representing the cost, distance, or capacity.

Graph Algorithms

Graph algorithms are fundamental in solving various optimization problems[35]–[39]:

1) Shortest Path: Algorithms like Dijkstra's and A* are used to find the shortest path between nodes in a weighted graph.
   Dijkstra's Algorithm: Utilized for graphs with non-negative weights to find the shortest path from a source node to all other nodes.
   $$Initialization: d(v) = \infty \forall v \in V, d(source) = 0 \quad (1)$$
   $$Relaxation: d(v) = \min\left(d(v), d(v) + w(u, v)\right) \forall\ (u, v) \in E \quad (2)$$
2) Max Flow/Min Cut: Algorithms like Ford-Fulkerson are used to find the maximum flow in a network.

Ford-Fulkerson Algorithm: Determines the maximum flow from a source to a sink in a flow network.

$Residual\ Capacity: cf(u,v) = c(u,v) - f(u,v)$ (3)

$Augment\ Flow: f(u,v) = f(u,v) + \Delta f$ (4)

3) Graph Traversal: Algorithms like Depth-First Search (DFS) and Breadth-First Search (BFS) explore the nodes of a graph.

Depth-First Search (DFS): Explores as far along a branch as possible before backtracking.

DFS Visit: Visit $(u) \rightarrow$ for each $v \in$ Adj $(u), if\ v$ is unvisited, DFS (v) (5)

Breadth-First Search (BFS): Explores all neighbors of a node before moving to the next level.

IBFS Initialization: $d(source) = 0, Q = \{source\}$ (6)

BFS Visit: for each $u \in Q$, for each $v \in Adj(u), if\ v$ is unvisited, $d(v) = d(u) + 1$ (7)

4) Community Detection: Algorithms such as Girvan-Newman detect clusters or communities within a graph.

Girvan-Newman Algorithm: Detects communities by iteratively removing edges with the highest betweenness centrality.

Edge Betweenness Centrality: $\sum_{s,t \in V} \frac{\sigma_{st}\ (e)}{\sigma_{st}}$, where σ_{st} (8)

is the total number of shortest paths from s to t and $\sigma_{st}(e)$

is the number of those paths that pass through edge e.

Optimization Techniques

Combining graph-based models with optimization techniques enhances their problem-solving capabilities[40]–[44]:

1) Linear Programming (LP):

Formulate optimization problems with linear constraints and objectives.

Objective Function: $min/\ max\ c^T x$ (9)

$Subject\ to: Ax \leq b, x \geq 0$

2) Genetic Algorithms (GA):

Use evolutionary techniques to find approximate solutions to optimization problems.

Initialization: Generate initial population

Selection: Select individuals based on fitness

Crossover: Combine pairs of individuals to produce offspring

Mutation: Introduce random changes

3) Machine Learning Integration:

Implement Graph Neural Networks (GNNs) to learn from graph-structured data.

Node Embeddings: $h_v^{(k)} = \sigma\left(W^k.\text{AGGREGATE}\left(\left\{h_u^{(k-1)}: u \in N(u)\right\}\right)\right)$ (10)

Use Reinforcement Learning for dynamic optimization.

Reward Function: $\text{R(s, a)} =$ immediate reward for taking action a in state s (11)

Policy Update: $\pi(\alpha|s) = \pi(\alpha|s) + \alpha\left(R(s,a) + \gamma_{a'}^{max} Q(s',a') - Q(s,a)\right)$ (12)

Proposed new method for Graph-based Exploration for Mining and Optimization of Yields (GEMOY Method)

To develop a new mathematical formulation for graph-based exploration aimed at optimizing yields in mining and other industries, we need to combine elements of graph theory, optimization algorithms, and machine learning. This formulation will focus on representing the complex systems involved, optimizing resource allocation, and improving overall efficiency.

Graph Representation

Define a graph $G = V, E)$ where:

- V is the set of vertices representing entities (e.g., mining sites, production units, agricultural fields).
- E is the set of edges representing relationships or interactions (e.g., transportation routes, dependencies).

Each edge $e \in E$ has an associated weight $w(e)$ which can represent cost, distance, or capacity.

Objective Function

Let x_i be the yield or output at node $i \in V$. The primary objective is to maximize the total yield while minimizing costs and other constraints.

$$Maximize\ Z = \sum_{i \in V} x_i - \sum_{e \in E} w(e).f(e) \tag{13}$$

Where $f(e)$ is the flow or utilization of edge e

Constraints

1) Resource Constraints:

$$\sum_{j \in \mathcal{N}(i)} f(i,j) \le R_i, \forall i \in V \tag{14}$$

Where $\mathcal{N}(i)$ is the set of neighbors of node i and R_i is the resource limit at node i.

2) Capacity Constraints:

$$0 \le f(e) \le c(e), \forall e \in E \tag{15}$$

Where $c(e)$ is the capacity of edge e.

3) Flow Conservation (for directed graphs):

$$\sum_{j \in \mathcal{N}^+(i)} f(i,j) - \sum_{j \in \mathcal{N}^-(i)} f(j,i) = b_i, \quad \forall i \in V \tag{16}$$

Where $\mathcal{N}^+(i)$ and $\mathcal{N}^-(i)$ are the sets of outgoing and incoming neighbors of node i, respectively, and b_i is the supply or demand at node i.

Graph-based Algorithms

1) Shortest Path with Yield Maximization:

$Minimize \sum_{e \in P} w(e)$

subject to:

P={path from source to destination}, (17)

$maximizing \sum_{i \in P} x_i$

2) Max Flow with Yield Optimization:

$Maximize \sum_{i \in V} x_i$

subject to: (18)

$\sum_{i \in E} f(e) \le \sum_{e \in E} c(e),\ f(e) \ge 0, \forall e \in E$

3) Node and Edge Weight Updates using Machine Learning:

Implement a Graph Neural Network (GNN) to predict yields and update weights:

$$h_v^{(k)} = \sigma\left(W^k.\text{AGGREGATE}\left(\{h_u^{(k-1)}: u \in \mathcal{N}(v)\}\right)\right) \tag{19}$$

Where $h_v^{(k)}$ is the feature vector of node v at layer k, W^k is the weight matrix, and σ is an activation function.

Optimization Technique Integration

1) Linear Programming (LP):

Formulate the yield optimization as a linear program:

$Maximize\ c^T x$

subject to: (20)

$Ax \le b, \quad x \ge 0$

2) Genetic Algorithms (GA):
Initialize a population of potential solutions.
Evaluate fitness based on the objective function Z.
Apply crossover and mutation operations to evolve the population toward optimal solutions.
3) Reinforcement Learning (RL):
Define a reward function $R(s,a)$ that incorporates yield optimization and resource costs. Update the policy $\pi(a|s)$ based on the reward feedback:

$$\pi(a|s) = \pi(a|s) + \alpha\left(R(s,a) + \gamma_{a'}^{max} Q(s',a') - Q(s,a)\right) \quad (21)$$

3. Results and Discussion

To test the GEMOY formulation in the previous section below let us discuss a detailed numerical example to optimise the yield in mining operations using graph-based methods.

Example Setup

We have the following components in a mining operation:
3 Mining Sites (Nodes): $\boldsymbol{A, B, C}$.
2 Processing Plants (Nodes): $\boldsymbol{P1, P2}$.
5 Transportation Routes (Edges): Connecting mining sites to processing plants.
The graph $\boldsymbol{G = (V, E)}$ is defined as:
$\boldsymbol{V = \{A, B, C, P1, P2\}}$
$\boldsymbol{E = \{(A, P1), (A, P2), (B, P1), (B, P2), (C, P2)\}}$
Each edge has an associated cost, and each node has a yield value.

Node Yields and Edge Costs
Node Yields:
A: 5 Units
B: 40 Units
C: 60 Units

Edge Costs:
(A,P1): 10
(A,P2): 20
(B,P1): 15
(B,P2): 25
(C,P2): 10

Objective

Maximize the total yield while minimizing the transportation costs.

Formulation

1) Objective Function:

$$Maximize\ Z = \sum_{i \in \{A,B,C\}} x_i - \sum_{e \in E} w(f).f(e)$$

Where x_i is the yield at node i, and $f(e)$ is the flow on edge e.
2) Constraints:
Flow conservation:

$$\sum_{j \in \mathcal{N}^+(i)} f(i,j) - \sum_{j \in \mathcal{N}^-(i)} f(i,j) = x_i\,, \forall i \in \{A,B,C\}$$

Capacity constraints (assuming no capacity limits for simplicity)

$$0 = f(e) \leq \infty, \forall e \in E$$

Solution

1) Graph Construction:
G=(V,E),V={A,B,C,P1,P2},E={(A,P1),(A,P2),(B,P1),(B,P2),(C,P2)}

2) Initial Flow Assignment:
   Assume initial flows based on yield distribution:
   f(A,P1)=30,f(A,P2)=20,f(B,P1)=20,f(B,P2)=20,f(C,P2)=60
3) Calculate Initial Cost:
   Total Cost=30·10+20·20+20·15+20·25+60·10=300+400+300+500+600=2100
4) Apply Optimization (e.g., Linear Programming):
   Formulate the LP problem to adjust flows for minimizing costs while maintaining yields:

$$Minimize \sum_{e \in E} w(e).f(e)$$

   subject to:

$$\sum_{j \in \mathcal{N}^+(i)} f(j,i) - \sum_{j \in \mathcal{N}^-(i)} f(j,i) = x_i \, , \forall i \in \{A,B,C\}$$

5) Optimization Using Linear Programming:
   Define the LP problem in matrix form:
   Decision Variables:

$$f(A,P1) = x_1, \quad f(A,P2) = x_2, \quad f(B,P1) = x_3, f(B,P2) = x_4, \quad f(C,P2) = x_5$$

   Objective Function:

$$Minimize\ 10x_1 + 20x_2 + 15x_3 + 25x_4 + 10x_5$$

   Constraints:

$$x_1 + x_2 = 50 \text{ (Yield at A)}$$
$$x_3 + x_4 = 40 \text{ (Yield at B)}$$
$$x_5 = 60 \text{ (Yield at C)}$$
$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

6) Solve the Linear Programming Problem:
   Using an LP solver, we find the optimal solution for the decision variables:

$$x_1 = 50, \quad x_2 = 0, \quad x_3 = 40, \quad x_4 = 0, \quad x_5 = 60$$

7) Calculate Optimized Cost:

$$Optimized\ Total\ Cost = 50 \cdot 10 + 0 \cdot 20 + 40 \cdot 15 + 0 \cdot 25 + 60 \cdot 10$$
$$= 500 + 0 + 600 + 0 + 600 = 1700$$

8) Result:
   The optimized solution yields a total cost of 1700 units with the following flows:

$$f(A,P1) = 50$$
$$f(B,P1) = 40$$
$$f(C,P2) = 60$$

By applying graph-based optimization techniques, we adjusted the flow distribution in the mining operation to minimize transportation costs while maintaining the yields. The total transportation cost was reduced from 2100 units to 1700 units. This example demonstrates the effectiveness of using advanced mathematical formulations and optimization algorithms in real-world industrial applications, providing a clear pathway to achieving operational efficiency and cost savings.

The numerical example presented demonstrates the application of graph-based optimization techniques to a mining operation. Initially, the total transportation cost for distributing yields from three mining sites to two processing plants was calculated to be 2100 units. By formulating the problem using linear programming, we aimed to minimize transportation costs while ensuring the yields from each mining site were appropriately allocated to the processing plants. The linear programming solution provided an optimized flow distribution: 50 units from Site A to Plant P1, 40 units from Site B to Plant P1, and 60 units from Site C to Plant P2. This optimized allocation reduced the total transportation cost to 1700 units.

The result highlights the effectiveness of using graph-based methods for yield optimization in a mining context. The initial arbitrary allocation resulted in higher transportation costs due to inefficient distribution of yields across the transportation routes. By leveraging the optimization techniques, significant cost savings were achieved, demonstrating the potential for substantial improvements in operational efficiency. The constraints ensured that each mining site's yield was fully utilized, and the optimal path selection minimized the transportation expenses.

This example underscores the practical applicability of graph theory and linear programming in industrial optimization problems. It illustrates how complex systems can be modeled and optimized using mathematical formulations, leading to tangible economic benefits. Moreover, the approach can be extended to more complex scenarios with additional constraints and larger networks, providing a robust framework for optimizing operations in various industries. The successful application in this simplified case sets the stage for further research and development of more sophisticated models and algorithms tailored to specific industrial contexts.

4. Conclusion

This research successfully demonstrates the application of graph-based optimization techniques to enhance yield management and minimize transportation costs in a mining operation. By representing the mining sites and processing plants as nodes and the transportation routes as edges in a graph, we formulated an optimization problem that maximizes yields while minimizing associated costs. The numerical example illustrated the substantial cost savings achieved through the use of linear programming, reducing transportation costs from 2100 units to 1700 units. The approach underscores the value of integrating graph theory, optimization algorithms, and machine learning to address complex industrial challenges. The ability to model interactions and dependencies in a graph framework enables more efficient resource allocation and operational planning. Furthermore, the flexibility of the method allows for adaptation to various industrial contexts, from mining to agriculture and beyond. This research paves the way for further exploration and refinement of graph-based optimization techniques. Future work could focus on incorporating additional constraints, such as capacity limits and dynamic changes in resource availability, to create even more robust models. Additionally, the integration of advanced machine learning algorithms, such as graph neural networks, can enhance the predictive capabilities and adaptability of the system. In conclusion, graph-based exploration and optimization provide a powerful toolkit for improving operational efficiency and achieving cost-effective resource management in diverse industrial applications. The demonstrated benefits in this study highlight the potential for broader adoption and development of these techniques, promising significant economic and operational advancements.

References

[1] S. Bag, M. S. Rahman, G. Srivastava, H.-L. Chan, and D. J. Bryde, "The role of big data and predictive analytics in developing a resilient supply chain network in the South African mining industry against extreme weather events," *Int. J. Prod. Econ.*, vol. 251, no. 9, p. 108541, 2022, doi: https://doi.org/10.1016/j.ijpe.2022.108541.

[2] M.-L. Tseng, T. P. T. Tran, H. M. Ha, T.-D. Bui, and M. K. Lim, "Sustainable industrial and operation engineering trends and challenges Toward Industry 4.0: A data driven analysis," *J. Ind. Prod. Eng.*, vol. 38, no. 8, pp. 581–598, 2021, doi: https://doi.org/10.1080/21681015.2021.1950227.

[3] M. Bedinger, L. Beevers, G. H. Walker, A. Visser-Quinn, and K. McClymont, "Urban systems: Mapping interdependencies and outcomes to support systems thinking," *Earth's Futur.*, vol. 8, no. 3, p. e2019EF001389, 2020, doi: https://doi.org/10.1029/2019EF001389.

[4] A. Veremyev, A. Semenov, E. L. Pasiliao, and V. Boginski, "Graph-based exploration and clustering analysis of semantic spaces," *Appl. Netw. Sci.*, vol. 4, no. 104, pp. 1–26, 2019, doi: https://doi.org/10.1007/s41109-019-0228-y.

[5] J. Devezas and S. Nunes, "A review of graph-based models for entity-oriented search," *SN Comput. Sci.*, vol. 2, no. 6, p. 437, 2021, doi: https://doi.org/10.1007/s42979-021-00828-w.

[6] J. Cao, Y. He, and Q. Zhu, "Feedstock scheduling optimization based on novel extensible p-graph reasoning in ethylene production," *Ind. Eng. Chem. Res.*, vol. 59, no. 42, pp. 18965–18976, 2020, doi:

https://doi.org/10.1021/acs.iecr.0c03563.

[7] M. Herrera, M. Pérez-Hernández, A. Kumar Parlikad, and J. Izquierdo, “Multi-agent systems and complex networks: Review and applications in systems engineering,” *Processes*, vol. 8, no. 3, p. 312, 2020, doi: https://doi.org/10.3390/pr8030312.

[8] S. E. Phule, “Graph Theory Applications in Database Management,” *Int. J. Sci. Res. Mod. Sci. Technol.*, vol. 3, no. 3, pp. 13–17, 2024, doi: ttps://doi.org/10.59828/ijsrmst.v3i3.190.

[9] J. L. Gross, J. Yellen, and M. Anderson, *Graph theory and its applications*. Chapman and Hall/CRC, 2018. doi: https://doi.org/10.1201/9780429425134.

[10] A. H. Osman and O. M. Barukub, “Graph-based text representation and matching: A review of the state of the art and future challenges,” *IEEE Access*, vol. 8, pp. 87562–87583, 2020, doi: 0.1109/ACCESS.2020.2993191.

[11] J. Gohil, J. Patel, J. Chopra, K. Chhaya, J. Taravia, and M. Shah, “Advent of Big Data technology in environment and water management sector,” *Environ. Sci. Pollut. Res.*, vol. 28, no. 45, pp. 64084–64102, 2021, doi: ttps://doi.org/10.1007/s11356-021-14017-y.

[12] E. Micheni, J. Machii, and J. Murumba, “Internet of things, big data analytics, and deep learning for sustainable precision agriculture,” in *2022 IST-Africa Conference (IST-Africa)*, IEEE, 2022, pp. 1–12. doi: 10.23919/IST-Africa56635.2022.9845510.

[13] M. K. Saggi and S. Jain, “A survey towards an integration of big data analytics to big insights for value-creation,” *Inf. Process. Manag.*, vol. 54, no. 5, pp. 758–790, 2018, doi: https://doi.org/10.1016/j.ipm.2018.01.010.

[14] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, and L. Petersson, “Graph-based deep learning for medical diagnosis and analysis: past, present and future,” *Sensors*, vol. 21, no. 14, p. 4758, 2021, doi: https://doi.org/10.3390/s21144758.

[15] L. Kotzur *et al.*, “A modeler’s guide to handle complexity in energy systems optimization,” *Adv. Appl. Energy*, vol. 4, no. 33, p. 100063, 2021, doi: https://doi.org/10.1016/j.adapen.2021.100063.

[16] S. Ren, Y. Zhang, Y. Liu, T. Sakao, D. Huisingh, and C. M. V. B. Almeida, “A comprehensive review of big data analytics throughout product lifecycle to support sustainable smart manufacturing: A framework, challenges and future research directions,” *J. Clean. Prod.*, vol. 210, no. 21, pp. 1343–1365, 2019, doi: https://doi.org/10.1016/j.jclepro.2018.11.025.

[17] J. Wang, W. Zhang, Y. Shi, S. Duan, and J. Liu, “Industrial big data analytics: challenges, methodologies, and applications,” *arXiv Prepr. arXiv1807.01016*, vol. 3, no. 7, pp. 365–400, 2018, doi: https://doi.org/10.48550/arXiv.1807.01016.

[18] R. Wu and L. Jiang, “Recovering dynamic networks in big static datasets,” *Phys. Rep.*, vol. 912, pp. 1–57, 2021, doi: https://doi.org/10.1016/j.physrep.2021.01.003.

[19] Y. Tian *et al.*, “Evolutionary large-scale multi-objective optimization: A survey,” *ACM Comput. Surv.*, vol. 54, no. 8, pp. 1–34, 2021, doi: https://doi.org/10.1145/3470971.

[20] S. R. Sinsel, R. L. Riemke, and V. H. Hoffmann, “Challenges and solution technologies for the integration of variable renewable energy sources—a review,” *Renew. energy*, vol. 145, no. 1, pp. 2271–2285, 2020, doi: https://doi.org/10.1016/j.renene.2019.06.147.

[21] J. Hester, T. R. Miller, J. Gregory, and R. Kirchain, “Actionable insights with less data: guiding early building design decisions with streamlined probabilistic life cycle assessment,” *Int. J. Life Cycle Assess.*, vol. 23, no. 23, pp. 1903–1915, 2018, doi: https://doi.org/10.1007/s11367-017-1431-7.

[22] J. Otala, A. Minard, G. Madraki, and S. Mousavian, “Graph-based modeling in shop scheduling problems: Review and extensions,” *Appl. Sci.*, vol. 11, no. 11, p. 4741, 2021, doi: https://doi.org/10.3390/app11114741.

[23] A. D. Pazho, G. A. Noghre, A. A. Purkayastha, J. Vempati, O. Martin, and H. Tabkhi, “A survey of graph-based deep learning for anomaly detection in distributed systems,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 1–20, 2023, doi: https://doi.org/10.1109/TKDE.2023.3282898.

[24] C. Musto, P. Lops, M. de Gemmis, and G. Semeraro, "Context-aware graph-based recommendations exploiting personalized pagerank," *Knowledge-Based Syst.*, vol. 216, no. 15, p. 106806, 2021, doi: https://doi.org/10.1016/j.knosys.2021.106806.

[25] T. M. Phuong and N. D. Phuong, "Graph-based context-aware collaborative filtering," *Expert Syst. Appl.*, vol. 126, no. 15, pp. 9–19, 2019, doi: https://doi.org/10.1016/j.eswa.2019.02.015.

[26] E. Skakalina, "Application of ant optimization algorithms in the solution of the routing problem," *Системи управління, навігації та зв'язку. Збірник наукових праць*, vol. 6, no. 58, pp. 75–83, 2019, doi: https://doi.org/10.26906/SUNZ.2019.6.075.

[27] S. Abadal, A. Jain, R. Guirado, J. López-Alonso, and E. Alarcón, "Computing graph neural networks: A survey from algorithms to accelerators," *ACM Comput. Surv.*, vol. 54, no. 9, pp. 1–38, 2021, doi: https://doi.org/10.1145/3477141.

[28] Q. Cappart, D. Chételat, E. B. Khalil, A. Lodi, C. Morris, and P. Veličković, "Combinatorial optimization and reasoning with graph neural networks," *J. Mach. Learn. Res.*, vol. 24, no. 130, pp. 1–61, 2023, [Online]. Available: https://www.jmlr.org/papers/v24/21-0449.html

[29] V.-A. Darvariu, S. Hailes, and M. Musolesi, "Graph Neural Modeling of Network Flows," 2022. doi: https://doi.org/10.48550/arXiv.2209.05208.

[30] S. Thurner, R. Hanel, and P. Klimek, *Introduction to the theory of complex systems*. USA: Oxford University Press, 2018.

[31] A. Majeed and I. Rauf, "Graph theory: A comprehensive survey about graph theory applications in computer science and social networks," *Inventions*, vol. 5, no. 1, p. 10, 2020, doi: https://doi.org/10.3390/inventions5010010.

[32] J. Jalving, Y. Cao, and V. M. Zavala, "Graph-based modeling and simulation of complex systems," *Comput. Chem. Eng.*, vol. 125, no. 45, pp. 134–154, 2019, doi: https://doi.org/10.1016/j.compchemeng.2019.03.009.

[33] I. van Gent and G. La Rocca, "Formulation and integration of MDAO systems for collaborative design: A graph-based methodological approach," *Aerosp. Sci. Technol.*, vol. 90, no. 7, pp. 410–433, 2019, doi: https://doi.org/10.1016/j.ast.2019.04.039.

[34] A. Laurent *et al.*, "Methodological review and detailed guidance for the life cycle interpretation phase," *J. Ind. Ecol.*, vol. 24, no. 5, pp. 986–1003, 2020, doi: https://doi.org/10.1111/jiec.13012.

[35] D. Jungnickel and D. Jungnickel, *Graphs, networks and algorithms*, vol. 3. Springer, 2005. doi: https://doi.org/10.1007/978-3-540-72780-4.

[36] J. van Leeuwen, "Graph algorithms," in *Algorithms and complexity*, Elsevier, 1990, pp. 525–631. doi: https://doi.org/10.1016/B978-0-444-88071-0.50015-1.

[37] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song, *Learning combinatorial optimization algorithms over graphs*, vol. 30. 2017.

[38] N. Andréasson, A. Evgrafov, and M. Patriksson, *An introduction to optimization: Foundations and fundamental algorithms*, vol. 1. 2005. [Online]. Available: https://www.math.chalmers.se/Math/Grundutb/CTH/tma947/0405/kompendium-2.pdf

[39] S. Even, *Graph algorithms*. Cambridge University Press, 2011.

[40] R. C. Holte, T. Mkadmi, R. M. Zimmer, and A. J. MacDonald, "Speeding up problem solving by abstraction: A graph oriented approach," *Artif. Intell.*, vol. 85, no. 1–2, pp. 321–361, 1996, doi: https://doi.org/10.1016/0004-3702(95)00111-5.

[41] R. Marinescu and R. Dechter, "AND/OR branch-and-bound search for combinatorial optimization in graphical models," *Artif. Intell.*, vol. 173, no. 16–17, pp. 1457–1491, 2009, doi: https://doi.org/10.1016/j.artint.2009.07.003.

[42] X. Zhu and J. Lafferty, "Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 1052–1059. doi: https://doi.org/10.1145/1102351.1102484.

[43] A. L. Bertozzi and E. Merkurjev, “Graph-based optimization approaches for machine learning, uncertainty quantification and networks,” in *Handbook of Numerical Analysis*, vol. 20, Elsevier, 2019, pp. 503–531. doi: https://doi.org/10.1016/bs.hna.2019.04.001.

[44] Y. Peng, B. Choi, and J. Xu, “Graph learning for combinatorial optimization: a survey of state-of-the-art,” *Data Sci. Eng.*, vol. 6, no. 2, pp. 119–141, 2021, doi: https://doi.org/10.1007/s41019-021-00155-3.