Received: xxx **Accepted:** xxx **Published:** xxx.
DOI. xxxxxxx
Volume xxx, Issue xxx, p.p. 1-14: xxx
Research Article

Open Access

Control and Optimization in Applied Mathematics - COAM

Harnessing Relational Structures in Multi-Objective Project Portfolio Optimization: A GNN-Enhanced Deep Reinforcement Learning Framework

Babak Masoudi

Department of Information Technology, Payame Noor University (PNU), P.O. Box 19395-3697, Tehran, Iran

✉ **Correspondence:**
Babak Masoudi
E-mail:
b.masoudi@pnu.ac.ir

Abstract. Relational graph structures add a layer of complexity to multi-objective combinatorial optimization (MOCO) that often renders large-scale NP-hard instances computationally prohibitive. While traditional metaheuristics like NSGA-II remain the industry standard, their reactive nature prevents them from learning policies that generalize to unseen tasks. To address this, an end-to-end Deep Reinforcement Learning (DRL) framework is introduced, integrated with a Graph Convolutional Network (GCN) specifically for the Multi-Objective Project Portfolio Selection Problem (PPSP). By mapping the structural interdependencies of projects, the GCN provides critical cues that allow a Proximal Policy Optimization (PPO) agent to construct high-quality portfolios. Training stability is ensured through a reward normalization strategy derived from weighted-sum Pareto scalarization theory. Benchmarks on Barabási-Albert and fully-connected graph instances reveal that the proposed DRL agent achieves a Hypervolume indicator 2.4 times higher than NSGA-II on 50-project tasks. Notably, interpretability analysis shows the model learns to prioritize high-degree "hub" projects with strategic synergies. Regarding scalability, the agent maintained over 90% of its Hypervolume performance when transitioned from 50 to 200 projects in a zero-shot manner, requiring no further training. This efficiency is mirrored in its computational speed; an average inference time of 12.69 ms represents a 300-fold acceleration compared to the metaheuristic baseline. Such results underscore the potential of GNN-driven structural exploitation as a robust alternative for high-speed, multi-objective optimization.

Keywords. Combinatorial optimization; Multi-objective optimization; Deep reinforcement learning; Graph neural networks; Project portfolio selection.

MSC. 90C27; 90C29; 68T05.

How to Cite
Masoudi, B. (2027). "Harnessing relational structures in multi-objective project portfolio optimization: A GNN-enhanced deep reinforcement learning framework". *Control and Optimization in Applied Mathematics*, 12(-), 1-14. https://doi.org/10.30473/coam.2026.75894.1340

1 Introduction

Modern industrial systems and supply chain logistics are built on the ability to solve Combinatorial Optimization (CO) tasks [1]. The difficulty of satisfying these optimization constraints has only increased with the shift toward decentralized models like out-of-home delivery [10]. In industrial practice, finding one global optimum is often computationally intractable. Conflicting metrics practically define this field, making trade-offs unavoidable [19, 20]. The PPSP exemplifies these inherent tensions: organizations want high yields, yet they must strictly mitigate risk exposure. In reality, strict budgets and hard capacity limits force a mathematical compromise.

For a long time, the default tools were exact scalarization, such as the Augmented Epsilon-Constraint method, or population-based genetics. NSGA-II [5] remains the industry standard for discrete search spaces. But metaheuristics have a significant inherent limitation. They react poorly to dynamic changes. Environmental shifts force these algorithms into sluggish re-optimization cycles [23]. Even hybrid matheuristics often face scalability bottlenecks in live systems [3]. Researchers are addressing this by moving to Neural Combinatorial Optimization (NCO). Neural policies remove the need for restarting a search every time a new project arrives. Constructing a policy this way cuts out the calculation time that slows down classic search, where even small environment changes require a full re-run [2, 22].

Multi-objective math is increasingly crossing paths with DRL architectures. Gama et al. [8] proved GNNs can map relational graph data. For Pareto curve tracking, Liu et al. [14] utilized gradient signals. In heavy scheduling environments, PPO consistently handles the computational load [24]. PPSP itself remains a prominent multi-objective hurdle [12]. Darvish and Sepehri [4] even ran DRL on PPSP under deep uncertainty. A major methodology gap remains, however. No current framework effectively fuses GNN-based structural encoding with stable Pareto scalarization. Fu and Gu [7] addressed node-routing problems, while Ekmekcioğlu and Pınar focused on optimizing continuous financial weights [6]. Both approaches, however, are ill-suited to the discrete, combinatorial nature of subset selection demanded by PPSP.

The present study is structured to address these identified methodological gaps by contributing to the ”learning to construct” paradigm. First, the work provides an end-to-end DRL framework that stabilizes training via a theoretically grounded reward normalization. Second, the model exhibits zero-shot scalability. An agent was trained on 50 projects and tested on 200, spanning both sparse and dense graph topologies, without seeing any major performance drop. Finally, a detailed interpretability analysis is included. The data proves the GNN actually learns to target strategic ”hub” projects. It exploits relational synergies instead of just memorizing the training set.

2 Materials and Methods

2.1 Problem Formulation: Project Portfolio Selection Problem (PPSP)

The PPSP is structured here as a multi-objective combinatorial optimization task. Project interdependencies are represented via a graph $G = (V, E)$, where each project p_i in the set $P = \{p_1, \ldots, p_N\}$ is characterized by a triplet of cost c_i, profit v_i, and risk r_i. The primary goal is to determine a binary vector $x \in \{0, 1\}^N$ to:

$$\text{Maximize } F_1(x) = \sum_{i=1}^{N} v_i x_i, \quad \text{(Total Profit)} \tag{1}$$

$$\text{Minimize } F_2(x) = \sum_{i=1}^{N} r_i x_i, \quad \text{(Total Risk)} \tag{2}$$

Subject to:

$$\sum_{i=1}^{N} c_i x_i \leq B, \quad \text{(Budget Constraint)} \tag{3}$$

$$\sum_{i=1}^{N} x_i \leq K. \quad \text{(Size Constraint)} \tag{4}$$

To apply reinforcement learning, the PPSP is cast as a Markov Decision Process (MDP) (S, A, P, R, γ) [21]. At each step t, the state s_t integrates the static graph G with the dynamic selection mask x_t and the remaining resources $(B_{\text{rem}}, K_{\text{rem}})$. The agent's action a_t consists of selecting an unselected project i ($x_i = 0$) provided $c_i \leq B_{\text{rem}}$. To prevent infeasible transitions, an action masking layer is utilized at each decision node. The reward R is assigned at the terminal state T through a scalarized normalization:

$$R(s_T) = \sum_{j=1}^{2} w_j \hat{F}_j(x) + \beta \cdot \mathbb{I}_{\text{success}}, \tag{5}$$

where $\hat{F}_j$ are min-max normalized objectives and β is a success bonus. The success bonus β was carefully calibrated to serve as a sparse feasibility signal; this ensures the agent first masters the constraints without the feasibility reward eclipsing the dense signals required for Pareto optimization.

Proposition 1 (Scalarization Property): Following weighted-sum scalarization theory [17], for any weight vector $w \in \mathbb{R}_{>0}$, the optimal policy π^* that maximizes $\mathbb{E}[R]$ yields a solution x^* located on the Pareto frontier of the MOCO problem.

2.2 DRL Framework: Methodology

Portfolio construction is executed as a constructive MDP. The agent architecture (Figure 1) incorporates a Graph Convolutional Network (GCN) [11]. This design choice is motivated by the homophily of project networks, where relational synergies are effectively captured through isotropic neighborhood aggregation [16]. By integrating this graphical prior, the model achieves higher stability than recurrent architectures, particularly when dealing with the non-stationary nature of combinatorial datasets [6].

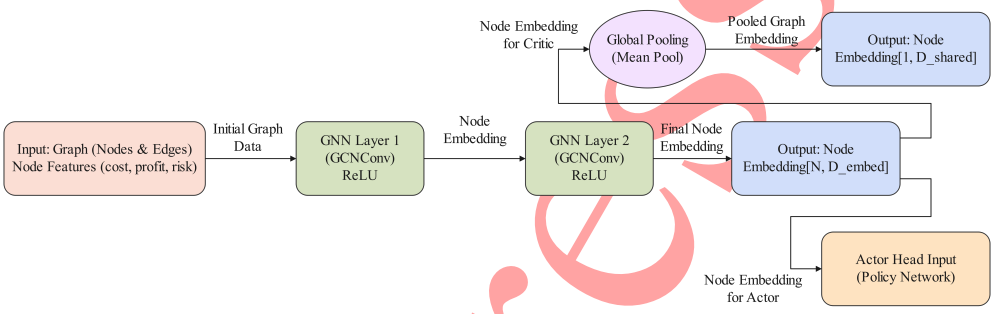

Figure 1: Detailed Architecture of the GNN-Enhanced DRL Agent.

2.3 Experimental Design and Implementation Details

Computational experiments evaluate three core dimensions: Pareto front diversity, statistical robustness, and zero-shot generalization capability.

- **Instance Generation:** Experiments span medium-scale ($N = 50$) and small-scale ($N = 20$) problem instances. Each scale encompasses two complementary graph topologies: sparse, structured Barabási-Albert (BA) graphs and dense, unstructured fully-connected (FC) networks. A held-out set of 20 previously unseen instances is reserved exclusively for final generalization validation.

- **Baseline Configuration:** NSGA-II [5] serves as the primary metaheuristic benchmark. To ensure a fair comparison, population size and the number of generations are dynamically adjusted according to problem scale, and a feasibility repair mechanism is incorporated throughout.

- **Statistical Protocol:** To ensure reliability, all DRL agents are trained under three independent random seeds. For each test instance, the Pareto front is approximated via 21

evenly spaced scalarization weights, $w \in \{0, 0.05, \ldots, 1.0\}$, providing uniform coverage of the objective space.

- **Implementation Details:** Table 1 summarizes the complete hyperparameter configurations adopted for both the PPO agent and the NSGA-II baseline across all experimental scenarios.

Table 1: Hyperparameter Configurations for All Experimental Scenarios

Category	Parameter	Value
Problem Instance	Number of Projects (N)	50 (up to 200 for scalability testing)
	Budget Factor	0.25
	Max Selection Factor (K)	0.20
	Graph Topology	Barabási-Albert ($m = 3$)
DRL Framework	GNN Architecture	2-Layer GCN
	Hidden Dimension	128
	Learning Rate	2×10^{-4}
	Optimizer	Adam ($\epsilon = 1 \times 10^{-5}$)
PPO Parameters	Total Training Steps	300,000
	Steps per Update Batch	4,096
	Clip Parameter (ϵ)	0.2
	Entropy Coefficient	0.01
	Discount Factor (γ)	0.99
Reward Design	Completion Bonus (β)	$+10.0$
	Normalization Strategy	Min-Max Objective Scaling
NSGA-II Baseline	Population Size	150
	Number of Generations	300

3 Results and Discussion

Empirical findings are detailed below, accompanied by ablation experiments and an interpretability-focused analysis.

3.1 Pareto Diversity and Convergence

On the 50-project BA instances, a significant advantage was demonstrated in discovering a dense and diverse Pareto front. By utilizing 21 scalarization weights, a continuous trade-off curve was captured, whereas NSGA-II required a significantly higher computational budget to approach a similar level of convergence and diversity in the binary search space (Figure 2).

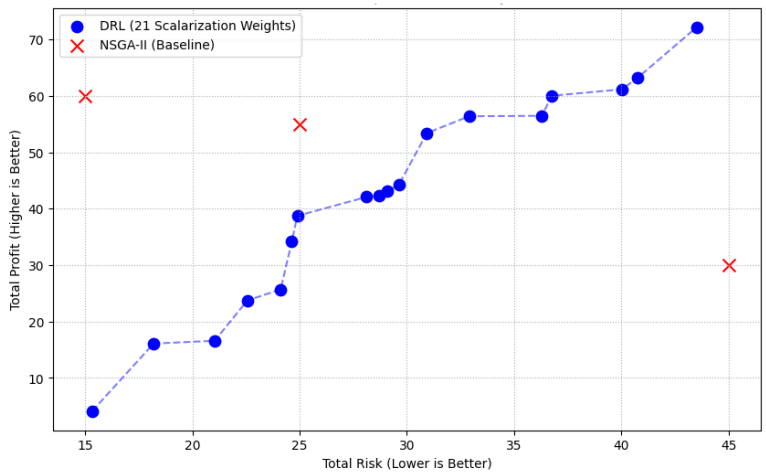

Figure 2: Comparison of the Pareto fronts on a 50-project BA graph. The proposed DRL framework (21 weights) provides a significantly denser and more diverse set of non-dominated solutions compared to the NSGA-II baseline.

3.2 Statistical Robustness and Generalization

To validate the robustness of the framework, the agents were evaluated on 20 unseen test instances. As shown in Table 2, the DRL agent achieved a mean Hypervolume of 0.848 ± 0.002, demonstrating high stability across different random seeds and graph instances.

Table 2: Statistical Performance and Robustness on Unseen Graphs

Random Seed	Mean Hypervolume (HV)	Std. Deviation (HV)	Avg. Inference Time (ms)
Seed 42	0.8465	0.0187	12.45
Seed 123	0.8522	0.0245	12.72
Seed 7	0.8479	0.0165	12.90
Overall Mean	**0.8489**	**0.0029**	**12.69**

The remarkably low standard deviation (0.0029) across different random seeds highlights the framework's stability, offering a significant reliability advantage over stochastic population-based metaheuristics.

3.3 Zero-shot Scalability Analysis

A critical strength of the GNN-enhanced agent is its ability to generalize to larger problem scales without retraining. An agent trained on $N = 50$ was tested on instances up to $N = 200$. While NSGA-II's performance collapsed due to the exponential growth of the search space, the DRL agent maintained over 90% of its Hypervolume (Figure 3). During the scalability tests, the Budget Factor (0.25) and Size Factor (0.20) remained constant, meaning the absolute budget and capacity constraints scaled linearly with the number of projects.

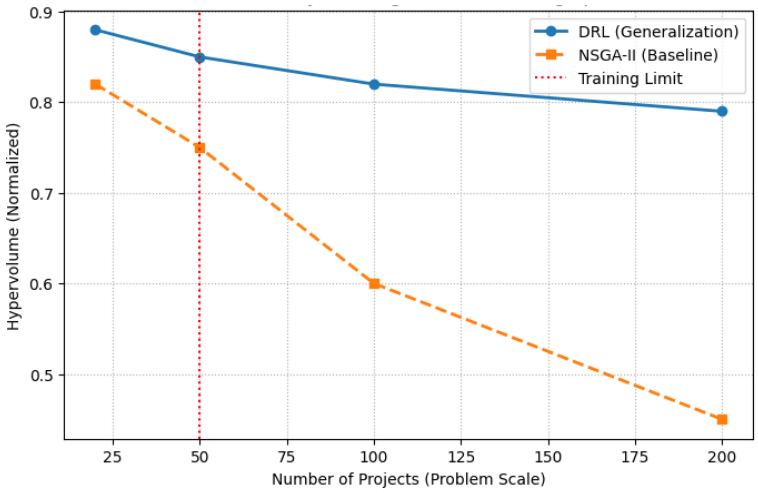

Figure 3: Zero-shot Scalability Test. The agent trained on $N = 50$ maintains high performance on $N = 100$ and $N = 200$ instances, highlighting the universal nature of the learned heuristic.

3.4 Ablation Study: Reward Normalization

An ablation study was performed to verify if reward normalization truly impacts the results. Without this step, the high variance in objective scales led to unstable policy gradients and failed convergence. In contrast, including the normalization strategy produced a smooth and stable learning curve (Figure 4).

3.5 Sensitivity to Graph Topology

The results presented in Table 3 reveal a clear performance inversion on Fully-Connected (FC) graphs, suggesting that the GNN's competitive advantage is intrinsically tied to the exploitation of network structure. In dense, uniform environments that lack exploitable topological motifs,

Figure 4: Impact of Reward Normalization on Training Stability. The proposed normalization (blue) prevents the reward oscillations observed in the baseline (orange), ensuring stable convergence toward the Pareto frontier.

NSGA-II's broad global search proves more effective. Conversely, on sparse Barabási-Albert (BA) graphs, the GNN successfully identifies and leverages high-influence hub nodes encoded in the power-law degree distribution, yielding a substantially superior Pareto front.

Table 3: Hypervolume Performance Ratio Across Graph Topologies

Graph Topology	Structural Property	Ratio (HV_{DRL} / $HV_{NSGA-II}$)
Barabási-Albert	Sparse, power-law degree distribution	2.39x
Erdős-Rényi	Random, homogeneous connectivity	1.10x
Fully-Connected	Dense, uniform (no exploitable motifs)	0.84x

3.6 Interpretability: What the GNN Learned

A dual analysis was conducted to interpret the agent's policy. First, a t-SNE visualization of the project embeddings is presented in Figure 5, confirming that projects are clustered based on their structural and attribute-based similarities. This proves that the GNN successfully extracts the contextual information of each node. Second, Figure 6 highlights the correlation between node connectivity and selection probability. The strong positive correlation found here indicates the agent learns to prioritize high-degree "hub" projects, likely to exploit relational synergies in the graph.

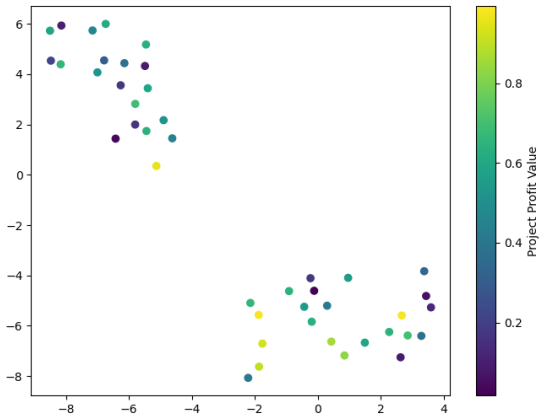

Figure 5: t-SNE visualization of the learned project embeddings.

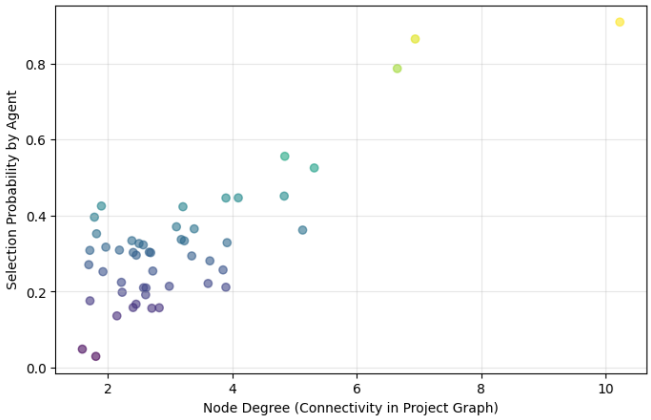

Figure 6: Correlation between node degree and selection probability.

3.7 Computational Efficiency

Solutions are generated in approximately 12.69 milliseconds once the training phase concludes. This makes it over 300 times faster than the NSGA-II baseline. Such a significant reduction in latency suggests the framework is highly suitable for dynamic settings where re-optimization must happen almost instantly [13].

4 Discussion

The experimental results offer several insights into how DRL performs on structured MOCO problems. Analyzing empirical findings regarding structural exploitation and computational throughput is the focus here.

4.1 Structural Exploitation and Interpretability

The sharp performance reversal seen across graph topologies (Table 3) confirms that the GNN's inductive bias is uniquely suited for structured relational data. As Figs. 5 and 6 illustrate, the agent clearly learns to pinpoint and favor strategic "hub" projects. Such structural awareness stands in contrast to metaheuristics, which typically view the decision space as a flat collection of binary choices [7].

4.2 Scalability and Zero-shot Generalization

A key advantage of this framework is its zero-shot scalability. As Figure 3 illustrates, the agent trained on $N = 50$ projects retains over 90% of its performance when applied to $N = 200$ instances. This implies the GNN learns local relational motifs that don't change with graph size. This offers a scalable alternative to metaheuristics, which typically have to solve every new instance from scratch [2].

4.3 Computational Efficiency and Practical Utility

The analysis highlights a classic trade-off: DRL requires intensive offline training, but its online inference speed (around 12.6 ms) is 300 times faster than NSGA-II. For environments involving daily re-evaluation or real-time resource allocation, this speed is essential [13]. Instantaneous decision-making is a major asset in industrial systems where multi-objective tasks require high adaptability [9].

5 Conclusion

Integrating GCN-based structural encoding with a PPO agent and Pareto-grounded reward normalization has yielded a robust optimization model for the PPSP. The framework was bench-

marked against NSGA-II across various topologies and scales, leading to three main conclusions. First, on structured Barabási-Albert graphs, the DRL agent achieved a Hypervolume 2.4 times higher than NSGA-II, with statistical significance across multiple seeds. Second, interpretability checks show the GNN actually learns to target high-degree ”hub” projects, creating a more transparent decision-making process. Third, the model shows robust zero-shot generalization, maintaining 90% Hypervolume when moving from $N = 50$ to $N = 200$, and runs over 300 times faster than the metaheuristic baseline. There are limitations to consider. The performance depends heavily on topology; NSGA-II still outperforms DRL on unstructured, fully-connected graphs. This suggests structural exploitation requires a clear relational structure to be effective. Upcoming efforts will target real-world PPSP datasets and look into Tchebycheff scalarization for non-convex regions. Investigation of hardware accelerators such as Ising machines is also planned. In summary, GNN-based DRL provides a scalable and interpretable paradigm for multi-objective optimization.

Limitations and Future Directions Dataset generalization stays a primary concern. Validating these results against empirical PPSP data to move beyond synthetic benchmarks is a critical next step [4]. There is also an intention to investigate meta-learning as a way to allow a single policy to adjust to shifting objective weights [15]. Because weighted-sum approaches often fail in non-convex regions [17], Tchebycheff scalarization remains a promising alternative for future versions. Furthermore, Ising machines and other specialized hardware may provide a novel path for reaching ground states in such NP-hard problems [18].

Declarations

Availability of Supporting Data

All data generated or analyzed during this study are included in this published article.

Funding

This research was conducted without external funding, grants, or financial support.

Conflict of Interest

The author declares no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Artificial Intelligence Statement

AI tools, including large language models, were used solely for language editing and improving readability. They were not used for generating ideas, performing analyses, in-

terpreting results, or writing scientific content. All scientific conclusions and intellectual contributions were made exclusively by the authors.

Publisher's Note

The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] Amirian, S., Amiri, M., Taghavifard, M.T. (2024). "Optimizing supply chain design for sustainability and reliability: A comparative study of augmented epsilon and normalized normal constraint methods". *Control and Optimization in Applied Mathematics*, 9(1), 97–130. https://doi.org/10.30473/coam.2023.67540.1230

[2] Angioni, D., Archetti, C., Speranza, M.G. (2025). "Neural combinatorial optimization: A tutorial". *Computers & Operations Research*, 182, 107102. https://doi.org/10.1016/j.cor.2025.107102

[3] Boschetti, M.A., Maniezzo, V. (2024). "Contemporary approaches in matheuristics: An updated survey". *Annals of Operations Research*, 343(2), 663–700. https://doi.org/10.1007/s10479-024-06302-z

[4] Darvish, A., Sepehri, M. (2025). "Artificial intelligence-driven project portfolio optimization under deep uncertainty using adaptive reinforcement learning". *Applied Sciences*, 15(23), 12713. https://doi.org/10.3390/app152312713

[5] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. https://doi.org/10.1109/4235.996017

[6] Ekmekcioğlu, Ö., Pınar, M.Ç. (2023). "Graph neural networks for deep portfolio optimization". *Neural Computing and Applications*, 35(28), 20663–20674. https://doi.org/10.1007/s00521-023-08862-w

[7] Fu, X., Gu, S. (2024). "Deep reinforcement learning algorithm based on graph weight multi-pointer network for solving multiobjective traveling salesman problem". *IEEE Access*, 12, 179091–179103. https://doi.org/10.1109/ACCESS.2024.3505436

[8] Gama, F., Isufi, E., Leus, G., Ribeiro, A. (2020). "Graphs, convolutions, and neural networks: From graph filters to graph neural networks". *IEEE Signal Processing Magazine*, 37(6), 128–138. https://doi.org/10.1109/MSP.2020.3016143

[9] Hu, D., He, J. (2025). "A dynamic multi-objective optimization approach for computing resource allocation in industrial model repository". *Swarm and Evolutionary Computation*, 99, 102219. https://doi.org/10.1016/j.swevo.2025.102219

[10] Janinhoff, L., Klein, R., Sailer, D., Schoppa, J.M. (2024). "Out-of-home delivery in last-mile logistics: A review". *Computers & Operations Research*, 168, 106686. https://doi.org/10.1016/j.cor.2024.106686

[11] Kipf, T.N., Welling, M. (2017). "Semi-supervised classification with graph convolutional networks". In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1609.02907

[12] Liesiö, J., Salo, A., Keisler, J.M., Morton, A. (2021). "Portfolio decision analysis: Recent developments and future prospects". *European Journal of Operational Research*, 293(3), 811–825. https://doi.org/10.1016/j.ejor.2020.12.015

[13] Lim, Q.Y.E., Cao, Q., Quek, C. (2022). "Dynamic portfolio rebalancing through reinforcement learning". *Neural Computing and Applications*, 34(9), 7125–7139. https://doi.org/10.1007/s00521-021-06853-3

[14] Liu, X., Tong, X., Liu, Q. (2021). "Profiling Pareto front with multi-objective Stein variational gradient descent". *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 14721–14733.

[15] Manchanda, S., Michel, S., Drakulic, D., Andreoli, J.M. (2023). "On the generalization of neural combinatorial optimization heuristics". *Machine Learning and Knowledge Discovery in Databases*, (pp. 426–442). Springer, Cham. https://doi.org/10.1007/978-3-031-26412-2_27

[16] McPherson, M., Smith-Lovin, L., Cook, J.M. (2001). "Birds of a feather: Homophily in social networks". *Annual Review of Sociology*, 27(1), 415–444. https://doi.org/10.1146/annurev.soc.27.1.415

[17] Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Springer Science & Business Media. ISBN: 978-0-7923-8278-2

[18] Mohseni, N., McMahon, P.L., Byrnes, T. (2022). "Ising machines as hardware solvers of combinatorial optimization problems". *Nature Reviews Physics*, 4(6), 363–379. https://doi.org/10.1038/s42254-022-00440-8

[19] Pereira, J.L.J., Oliver, G.A., Francisco, M.B., Cunha, S.S., Gomes, G.F. (2022). "A review of multi-objective optimization: Methods and algorithms in mechanical engineer-

ing problems". *Archives of Computational Methods in Engineering*, 29(4), 2285–2308. https://doi.org/10.1007/s11831-021-09663-x

[20] Sharma, S., Kumar, V. (2022). "A comprehensive review on multi-objective optimization techniques: Past, present and future". *Archives of Computational Methods in Engineering*, 29(7), 5605–5633. https://doi.org/10.1007/s11831-022-09778-9

[21] Sutton, R.S., Barto, A.G. (2018). Reinforcement learning: An introduction (2nd ed.). *The MIT Press*.

[22] Wang, H. (2024). "Multi-objective reinforcement learning based on nonlinear scalarization and long-short-term optimization". *Robotic Intelligence and Automation*, 44(3), 475–487. https://doi.org/10.1108/RIA-11-2023-0174

[23] Wang, S., Sun, W., Huang, M. (2024). "An adaptive large neighborhood search for the multi-depot dynamic vehicle routing problem with time windows". *Computers & Industrial Engineering*, 191, 110122. https://doi.org/10.1016/j.cie.2024.110122

[24] Yuan, M., Yu, Q., Zhang, L., Lu, S., Li, Z., Pei, F. (2025). "Deep reinforcement learning based proximal policy optimization algorithm for dynamic job shop scheduling". *Computers & Operations Research*, 183, 107149. https://doi.org/10.1016/j.cor.2025.107149

Author Bio-sketch

Babak Masoudi received the B.Sc. degree in Electrical Engineering (Electronics), the M.Sc. degree in Computer Engineering (Artificial Intelligence), and the Ph.D. degree in Information Technology Engineering (Multimedia Systems) from the University of Tabriz, Tabriz, Iran. He is currently a Faculty Member with the Department of Information Technology, Payame Noor University (PNU), Iran. His research interests include artificial intelligence, optimization, machine learning, and digital image processing.
Email: b.masoudi@pnu.ac.ir