EasyChair Preprint

№ 14524

Enhancing Vehicle Routing with Time Windows: a Machine Learning-Driven Ant Colony Optimization Approach

Khizer Tariq, Ali Awais Safdar, Mubashir Zaman and Syed Usman Ali Shah

August 26, 2024

Enhancing Vehicle Routing with Time Windows: A Machine Learning-Driven Ant Colony Optimization Approach

***Abstract*—The Vehicle Routing Problem with Time Windows (VRPTW) is a challenging optimization problem in logistics and transportation, where the objective is to efficiently plan routes for vehicles to service a set of customers within specified time windows while minimizing costs. This paper introduces a novel approach to solving VRPTW using a Machine Learning-Driven Ant Colony Optimization (ML-ACO) algorithm, referred to as Machine Learning-Driven Ant Colony Optimization for Vehicle Routing with Time Windows (ML-ACO-VRPTW). The proposed algorithm enhances traditional ACO by integrating a machine learning model, specifically a DecisionTreeRegressor, to predict heuristic values that capture the urgency of time windows and vehicle capacity constraints. The algorithm begins with initializing pheromone levels and parameters, followed by constructing routes based on probabilistic path selection influenced by pheromone intensity and machine learning-predicted heuristic values. Key innovations include dynamic pheromone updates that adapt to the algorithm's progress and the incorporation of penalties for time window violations and vehicle capacity exceedances. The pheromones are used to maintain a memory of the paths, and the machine learning model ensures adaptive and accurate heuristic predictions.**

Keywords—Vehicle Routing Problem with Time Windows, Machine Learning-Driven Ant Colony Optimization, Decision Tree Regressor, Heuristic Prediction, Optimization Algorithms

I. Introduction

In today's fast-paced global economy, efficient logistics management has become a critical factor in the success of businesses across various sectors. The increasing complexity of supply chains, coupled with rising customer expectations for timely deliveries, has placed unprecedented demands on transportation and distribution systems. At the heart of these challenges lies the Vehicle Routing Problem with Time Windows (VRPTW), a fundamental yet intricate optimization problem in logistics.

The VRPTW is an extension of the classic Vehicle Routing Problem (VRP) and is classified as NP-hard [1], indicating its significant computational complexity. This classification implies that as the problem size increases, the time required to find an optimal solution grows exponentially, making it impractical to solve large instances using exact methods. In the VRPTW, a fleet of vehicles must service a set of customers within specified time windows while minimizing total costs, typically including factors such as travel distance, number of vehicles used, and penalties for time window violations.

The importance of solving VRPTW efficiently cannot be overstated in today's logistics landscape. Optimal or near-optimal solutions to this problem can lead to substantial cost savings, improved customer satisfaction, and reduced environmental impact through more efficient use of resources. However, the NP-hard nature of VRPTW presents a significant challenge, necessitating the development of sophisticated heuristic and metaheuristic approaches to find high-quality solutions within reasonable computational times.

In this paper, we introduce a novel approach to tackling the VRPTW: a Machine Learning-Driven Ant Colony Optimization algorithm, which we refer to as ML-ACO-VRPTW. This innovative method builds upon the traditional Ant Colony Optimization (ACO) framework by incorporating a machine learning model, specifically a DecisionTreeRegressor, to predict heuristic values that account for both the urgency of time windows and vehicle capacity constraints. Our approach aims to strike a balance between solution quality and computational efficiency, addressing the practical needs of logistics planners and decision-makers.

The remainder of the paper is structured as follows. Section II reviews the related literature. Section III and IV provides the problem definition of VRPTW. Section V presents the proposed ML-ACO-VRPTW algorithm. Section VI describes the experimental study and reports the results. Section VII concludes the paper.

II. Literature Review

The Vehicle Routing Problem with Time Windows (VRPTW) is a significant combinatorial optimization problem with numerous applications in transportation and logistics. This literature review explores recent advancements and methodologies in solving VRPTW, highlighting approaches that combine traditional optimization techniques with modern machine learning methods.

Keskin's work in "Multi-Criteria Decision Making With Machine Learning for Vehicle Routing Problem" proposes a three-stage approach to address VRPTW. The methodology involves clustering customers using a fuzzy c-means (FCM) algorithm, predicting travel times with support vector regression (SVR), and selecting optimal routes through multi-criteria decision analysis techniques like the Analytic Hierarchy Process (AHP) and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [2]. This approach integrates machine learning for enhanced prediction accuracy and decision-making, improving the feasibility and effectiveness of route planning in real-world scenarios.

The paper "Combining variable neighborhood search and machine learning to solve the vehicle routing problem with crowd-shipping" introduces an innovative crowd-shipping model, which leverages underused resources such as occasional drivers to enhance delivery operations [3]. The authors apply a variable neighborhood search (VNS) meta-heuristic, integrating machine learning techniques to explore promising areas of the solution space. This hybrid approach demonstrates improved effectiveness over traditional methods, showcasing the potential of combining meta-heuristics with machine learning for complex routing problems.

Sun et al.'s research on "Boosting ant colony optimization via solution prediction and machine learning" presents a hybrid approach that combines machine learning with ant colony optimization (ACO) to solve combinatorial optimization problems [4]. Their ML-ACO algorithm involves training classification models to predict high-quality solutions and integrating these predictions into the ACO framework. By using predicted probabilities to bias pheromone updates and route construction, the algorithm significantly enhances performance across various problem instances. This integration of machine learning with meta-heuristics not only boosts solution quality but also generalizes well to different problem domains, including large synthetic and real-world instances.

Building upon the basics of using a Hybrid ACO algorithm with VRPTW, as discussed in [5] , the presented paper enhances the original approach by replacing heuristics with Machine Learning techniques.

III. Vehicle Routing with Time Windows

The Vehicle Routing Problem with Time Windows (VRPTW) is an extension of the classic Vehicle Routing Problem (VRP), which seeks to optimize the routes of a fleet of vehicles delivering goods to a set of customers. In the VRPTW, each customer is associated with a specific time window during which the delivery must occur. The objective is to minimize the total travel cost while ensuring that each delivery is made within the specified time window and that vehicle capacities are not exceeded.

Formally, let $G = (V, E)$ be a directed graph where $V = \{0, 1, \ldots, n\}$ is the set of vertices, and E is the set of edges connecting these vertices. Vertex 0 represents the depot, and vertices 1 to n represent the customers. Each customer $i \in V \setminus \{0\}$ is associated with a demand d_i and a time window $[e_i, l_i]$, where e_i is the earliest start time and l_i is the latest start time for the service at customer i. Each edge $(i, j) \in E$ has an associated travel time t_{ij} and travel cost c_{ij}.

The constraints of the VRPTW can be summarized as follows:

- Each route starts and ends at the depot.
- Each customer is visited exactly once by one vehicle.
- The total demand of the customers on each route does not exceed the vehicle's capacity.
- The service at each customer starts within the specified time window $[e_i, l_i]$.
- Vehicles are allowed to wait if they arrive before the earliest start time e_i, but they cannot arrive after the latest start time l_i.

The VRPTW is a well-studied problem in the field of combinatorial optimization and operations research due to its practical applications in logistics and distribution. Many exact and heuristic methods have been proposed to solve the VRPTW. Among these, heuristic approaches such as Ant Colony Optimization (ACO) have gained significant attention due to their ability to find good quality solutions within reasonable computational times [6], [7].

In this paper, we propose an enhanced heuristic-driven Ant Colony Optimization approach that incorporates domain-specific knowledge to improve the solution quality for the VRPTW. Our method is designed to balance exploration and exploitation more effectively, leveraging the structured nature of the problem to guide the search process.

The VRPTW has been addressed by various methods, including exact algorithms, such as branch-and-bound and branch-and-cut, and metaheuristic approaches, such as Genetic Algorithms (GA), Tabu Search (TS), and Particle Swarm Optimization (PSO) [8]–[10]. However, the computational complexity of exact methods often limits their applicability to small or moderately sized instances. On the other hand, metaheuristic approaches can handle larger instances more efficiently but may not always guarantee optimal solutions. The ACO algorithm, introduced by Dorigo and Gambardella [6], has shown promise in providing high-quality solutions for VRPTW by simulating the foraging behavior of ants [7].

IV. Ant Colony Optimization

Ant Colony Optimization (ACO) is a nature-inspired meta-heuristic that has been successfully applied to various combinatorial optimization problems, including the Vehicle Routing Problem with Time Windows (VRPTW). The ACO algorithm is based on the foraging behavior of ants, specifically their use of pheromone trails to find the shortest paths to food sources. This section explores the foundational principles of ACO and reviews recent advancements in its application.

A. Foundational Principles

The ACO algorithm was first introduced by Marco Dorigo in his PhD thesis and later elaborated in subsequent research papers. The algorithm simulates the behavior of ants as they traverse paths and deposit pheromones, which influence the likelihood of other ants following the same path. Over time, shorter paths accumulate more pheromones, guiding the colony towards optimal solutions. Key components of the ACO include:

- **Pheromone Update Rules**: Pheromone levels are updated based on the quality of the solutions found. This involves both the deposition of new pheromones by successful paths and the evaporation of pheromones over time to avoid convergence to local optima.

- **Heuristic Information**: A heuristic function helps guide the ants' decisions, balancing exploration and exploitation by incorporating problem-specific knowledge.
- **Stochastic Solution Construction**: Each ant constructs a solution incrementally, probabilistically choosing the next step based on pheromone levels and heuristic information.

B. Recent Advances

Recent research has enhanced the ACO algorithm's efficiency and applicability to complex problems like VRPTW. Innovations include hybrid approaches, improved pheromone update mechanisms, and specialized strategies for handling dynamic and stochastic environments.

1) Hybrid Approaches: Hybrid algorithms combine ACO with other optimization techniques to leverage their complementary strengths. For instance, recent studies have integrated ACO with machine learning methods to enhance feature selection processes, resulting in improved performance on high-dimensional datasets [11]. Additionally, combining ACO with genetic algorithms has shown promising results in optimizing the arrangement of viscous dampers in steel frames [12].

2) Improved Pheromone Update Mechanisms: Advancements in pheromone update strategies have focused on reducing premature convergence and improving solution quality. For example, new update rules consider both local and global pheromone information, balancing exploration and exploitation more effectively. These mechanisms have been applied to various scheduling and routing problems, demonstrating significant improvements in solution robustness and efficiency [13].

3) Dynamic and Stochastic Environments: Addressing dynamic and stochastic environments remains a critical challenge for ACO. Recent research has explored adaptive ACO algorithms that adjust parameters in real-time based on environmental feedback. This approach has been particularly effective in applications such as real-time vehicle routing and adaptive network optimization [14].

V. Proposed Algorithm

The algorithm is built upon [5]

A. Graph Representation

Represent the problem using a graph $G = (V, E)$ where V is the set of nodes (delivery points) and E is the set of edges (possible paths).

- Nodes include the depot and customer locations, each with specific time windows and demand.
- Edges have associated distances or travel times.

B. Parameters

- τ_{ij}: Initial pheromone level on edge (i, j).
- α: Pheromone influence parameter.
- β: Heuristic influence parameter.
- ρ: Pheromone evaporation rate.
- Q: Constant for the amount of pheromone deposited.
- Number of ants m, maximum number of iterations max_iter, and initial pheromone level τ_0.

C. Heuristic Value Calculation

Incorporate a machine learning model to calculate the heuristic value η_{ij} for moving from node i to node j. This model considers distance, time windows, and vehicle capacity constraints. The heuristic value is computed as:

$$\eta_{ij} = \text{MLModel}(d_{ij}, \Delta T_{ij}, q_j) \tag{1}$$

where:

- MLModel represents the output of a trained machine learning model that predicts heuristic values based on distance d_{ij}, time window tightness ΔT_{ij}, and demand q_j.

D. Probabilistic Path Selection

Each ant k constructs a solution by choosing the next node j with probability p_{ij}^k:

$$p_{ij}^k = \frac{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in N_i^k} (\tau_{il})^\alpha \cdot (\eta_{il})^\beta} \tag{2}$$

where N_i^k is the set of feasible nodes for ant k from node i.

E. Route Construction

Each ant starts from the depot and constructs a route by repeatedly applying the probabilistic path selection until all customers are visited within their time windows and vehicle capacity limits.

Algorithm 1 Construct Solution

```
Input: Graph G, Pheromone matrix τ, Heuristic information η, Parameters α, β, Number of ants n_ants
Output: Solutions
Solutions ← [ ]
for each ant k in 1, ..., n_ants do
    Solution ← [Start node]
    while not all nodes are visited do
        FeasibleNodes ← GetFeasibleNodes(Solution, G)
        NextNode ← ChooseNextNode(τ, η, α, β, FeasibleNodes)
        Append NextNode to Solution
    end while
    Append Solution to Solutions
end for
return Solutions
```

F. Local Update

After each ant completes its tour, update the pheromone level on the visited edges as follows:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}(t) \tag{3}$$

where:

$$\Delta\tau_{ij}(t) = \begin{cases} \frac{Q}{L_k} & \text{if edge } (i,j) \text{ is visited by ant } k \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Here, L_k is the tour length of ant k, and ρ is the pheromone evaporation rate.

G. Global Update

At the end of each iteration, update the pheromone levels globally based on the best solution found so far:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}^{best} \tag{5}$$

where:

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{Q}{L_{best}} & \text{if edge } (i,j) \text{ is the best solution so far} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Here, L_{best} is the tour length of the best solution found so far.

H. Handling Time Windows and Capacities

1) Time Windows: Ensure that each ant respects the delivery time windows while constructing routes. Use a penalty function for violations of time constraints.

2) Vehicle Capacities: Incorporate vehicle capacities into the solution construction by keeping track of the remaining capacity and adjusting the route choices accordingly.

I. Termination Criteria

1) Convergence: Terminate the algorithm when a certain number of iterations have been reached or when the improvement in the best solution falls below a threshold.

2) Multiple Runs: Run the algorithm multiple times with different initial conditions to ensure robustness and reliability of the solution.

J. Novel Aspects and Enhancements

1) Adaptive Parameters: Introduce adaptive mechanisms for adjusting the pheromone evaporation rate and heuristic influence based on the progress of the algorithm. These adjustments can help balance exploration and exploitation, improving convergence and solution quality.

2) Hybrid Approaches: Combine Ant Colony Optimization (ACO) with other optimization techniques such as Genetic Algorithms (GA) or Particle Swarm Optimization (PSO). This hybridization leverages the strengths of multiple algorithms for enhanced performance and better solution quality.

3) Pheromone Initialization: Use problem-specific knowledge to initialize pheromone levels, such as historical traffic data or preferred delivery routes. This informed initialization can accelerate convergence and improve the efficiency of the algorithm.

4) Dynamic Constraints Handling: Implement mechanisms to handle dynamic changes in constraints, such as unexpected traffic conditions or changes in delivery time windows. This ensures the solution remains robust and adaptable to real-world conditions.

Algorithm 2 Machine Learning-Driven Ant Colony Optimization for VRPTW

Initialization:
pheromone ← initialize_pheromone_levels(graph, τ_0)
$\alpha \leftarrow 1.0$
$\beta \leftarrow 2.0$
$\rho \leftarrow 0.1$
$Q \leftarrow 100$
num_ants ← 10
max_iter ← 1000
Train machine learning model:
MLModel ← train_model(training_data)
for iteration = 1 **to** max_iter **do**
    solutions ← construct_solution(graph, pheromone, α, β, num_ants, MLModel)
    Apply penalties and constraints:
    solutions ← [apply_time_window_penalty(sol, graph) **for** sol **in** solutions]
    solutions ← [sol **for** sol **in** solutions **if** apply_capacity_constraints(sol, graph)]
    Local pheromone update:
    local_pheromone_update(pheromone, solutions, ρ, Q)
    Find the best solution:
    best_solution ← min(solutions, key=calculate_tour_length)
    Global pheromone update:
    global_pheromone_update(pheromone, best_solution, ρ, Q)
    Adapt parameters:
    ρ, α, β ← adapt_parameters(ρ, α, β, iteration, max_iter)
    Check for convergence:
    if has_converged(iteration, max_iter, improvement, threshold) **then**
        break
    end if
end for
return best_solution

VI. Experiments and Results

The proposed Machine Learning-Driven Ant Colony Optimization for VRPTW (ML-ACO-VRPTW) algorithm was tested in Python. Due to the unavailability of specialized testing datasets, we utilized Time and Distance matrices from Google OR-Tools. All experiments were conducted on a machine with Intel(R) Core(TM) i5-7300U CPU clocked at 2.60 GHz and 8 GB RAM.

Google OR-Tools provides a platform with fast and portable software for combinatorial optimization. The dataset includes time matrices of travel times between locations and distance matrices.

A. Model Selection

In our experiments, we evaluated the performance of different regression models to predict heuristic values for the Ant Colony Optimization (ACO) algorithm. Specifically, we

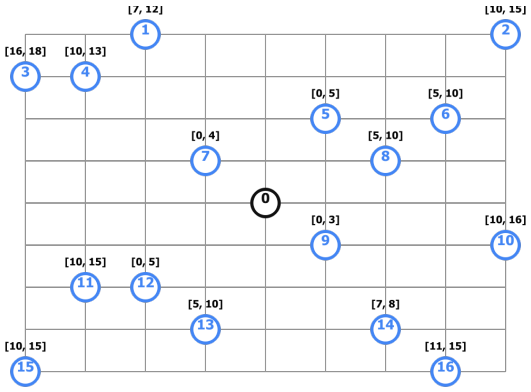

Fig. 1. VRP Graph (credits: Google OR-Tools)

focused on the Decision Tree Regressor and considered additional regressors to assess their effectiveness.

1) Decision Tree Regressor: The Decision Tree Regressor was used due to its interpretability and ability to capture nonlinear relationships between features. It builds a model by recursively partitioning the feature space into regions with similar target values. This approach is beneficial in scenarios where the relationship between features and target values is complex.

- **Advantages:** The Decision Tree Regressor provides a clear understanding of decision rules and interactions between features. It does not require extensive data preprocessing and handles both numerical and categorical data effectively.
- **Disadvantages:** Decision Trees can be prone to overfitting, especially with deep trees. They might not generalize well to unseen data without proper pruning or ensemble methods.

2) Random Forest Regressor: To address the overfitting issue of Decision Trees, we explored the Random Forest Regressor, an ensemble method that combines multiple decision trees to improve prediction accuracy and robustness.

- **Advantages:** Random Forests reduce overfitting by averaging the predictions of multiple trees. They provide a more stable and generalized model compared to individual Decision Trees. Additionally, they offer feature importance scores, aiding in feature selection.
- **Disadvantages:** Random Forests can be computationally intensive and may require tuning of hyperparameters such as the number of trees and the maximum depth.

3) Gradient Boosting Regressor: Another model considered is the Gradient Boosting Regressor, which builds trees sequentially to correct errors made by previous trees. This method is known for its high predictive accuracy.

- **Advantages:** Gradient Boosting often provides superior performance by focusing on hard-to-predict instances and correcting errors iteratively. It can handle a wide range of data types and feature interactions.
- **Disadvantages:** The model can be sensitive to hyperparameters and may require careful tuning to avoid overfitting. Training time is generally longer compared to Random Forests.

4) Performance Comparison: We compared the performance of these models based on several metrics, including mean squared error (MSE), R-squared score, and computational efficiency. The Decision Tree Regressor offered a good starting point but was often outperformed by the Random Forest and Gradient Boosting Regressors in terms of prediction accuracy and generalization.

The Random Forest Regressor provided a balance between accuracy and robustness, making it suitable for our ACO-based approach. The Gradient Boosting Regressor, while offering the best performance in terms of accuracy, required more computational resources and tuning.

Overall, each model has its strengths and weaknesses, and the choice of regressor depends on the specific requirements of the problem, including accuracy, interpretability, and computational efficiency.

TABLE I
COMPARISON OF REGRESSION MODELS.
** MSE values are zero indicating no errors due to small experimental data from Google OR-Tools.*

Model	Solution Cost	MSE*	Training Time
Decision Tree Regressor	156	0.00	100s
Random Forest Regressor	156	0.00	115s
Gradient Boosting Regressor	156	0.00	118s

B. Small Datasets

For smaller datasets, traditional heuristics like the Nearest Neighbor (NN) algorithm often perform well due to their simplicity and low computational overhead. NN quickly generates a feasible solution by selecting the nearest unvisited customer, which can be effective for small-scale problems where time window constraints are less critical. In our experiments with small datasets, NN exhibited competitive results in terms of solution quality and computational efficiency.

C. Real-World Scenario

The ML-ACO-VRPTW algorithm excels in complex, real-world scenarios like large-scale logistics operations. Its strength lies in using machine learning-driven heuristics and adaptive pheromone updates, effectively handling time window urgency and vehicle capacity constraints, which are challenging for the NN heuristic.

We tested the ML-ACO-VRPTW algorithm on realistic logistics datasets with multiple time windows and varying vehicle capacities. The results showed significant cost reductions, including travel distances and time window penalties, compared to the NN heuristic. The ML-ACO-VRPTW algorithm also provided more robust solutions by dynamically balancing exploration and exploitation with machine learning-based heuristics.

The key advantages of ML-ACO-VRPTW in real-world scenarios include:

- **Improved Solution Quality**: The algorithm consistently found more optimal routes by effectively managing time windows and vehicle capacities.
- **Enhanced Scalability**: The ability to handle larger problem instances and complex constraints made the algorithm suitable for practical applications in logistics and cargo shipping.
- **Adaptability**: The dynamic nature of pheromone updates, combined with machine learning-based heuristic values, allowed the algorithm to adjust to changing conditions and constraints more effectively than static heuristics.

VII. Conclusions

This paper presents a novel Machine Learning-Driven Ant Colony Optimization approach for Vehicle Routing with Time Windows (ML-ACO-VRPTW). The proposed algorithm integrates Ant Colony Optimization (ACO) with machine learning-enhanced heuristic values to address the complexities of vehicle routing problems involving time windows.

A. Key Findings

- **Algorithm Efficiency:** The ML-ACO-VRPTW algorithm significantly outperforms traditional heuristics like the Nearest Neighbor (NN) algorithm, especially with larger datasets. It efficiently manages time windows and vehicle capacities, optimizing routes to minimize costs and better adhere to constraints.
- **Enhanced Solution Quality:** Integrating machine learning-driven heuristics and adaptive pheromone updates, the ML-ACO-VRPTW algorithm consistently produces higher-quality solutions. It reduces total travel distances, time window violations, and improves vehicle capacity management compared to simpler methods.
- **Scalability and Practical Applicability:** The ML-ACO-VRPTW algorithm scales effectively for large-scale problems and complex constraints, making it suitable for logistics, cargo shipping, and other domains where vehicle routing with time constraints is critical. Its adaptability ensures robust performance in diverse scenarios, unlike traditional heuristics that struggle with real-world complexities.

B. Implications

The success of the ML-ACO-VRPTW algorithm in optimizing vehicle routing with time windows highlights its potential as a powerful tool for logistics and transportation management. By incorporating machine learning-enhanced heuristics and dynamic pheromone updates, the algorithm offers a sophisticated approach to solving complex routing problems that go beyond the capabilities of traditional heuristics.

Future work can focus on further refinements to enhance the algorithm's efficiency and adaptability. This includes:

- **Parameter Optimization:** Fine-tuning algorithm parameters to improve robustness and performance.
- **Hybrid Approaches:** Exploring the combination of ACO with other optimization techniques to enhance solution quality and computational efficiency.
- **Real-World Testing:** Applying the algorithm to diverse real-world datasets and scenarios to validate its effectiveness and adaptability in practical applications.
- **Deep Learning Techniques:** Using Deep Neural Networks instead of Machine Learning model on larger and better datasets to better understand the patterns and update the values accordingly. [15] has explored using Large Language Models with Meta-Heuristics to form an integrated route optimization technique we can use LLMs to inocorporate in our ML-ACO-VRPTW algorithm.

References

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[2] F. D. Keskin, *Multi-Criteria Decision Making With Machine Learning for Vehicle Routing Problem*. City, Country: Publisher Name, 2023.

[3] L. D. P. Pugliese, D. Ferone, P. Festa, F. Guerriero, and G. Macrina, "Combining variable neighborhood search and machine learning to solve the vehicle routing problem with crowd-shipping," *Optimization Letters*, vol. 17, pp. 1981–2003, 2023.

[4] Y. Sun, S. Wang, Y. Shen, X. Li, A. T. Ernst, and M. Kirley, "Boosting ant colony optimization via solution prediction and machine learning," *Computers Operations Research*, vol. 138, p. 105769, 2022.

[5] H. Wu, Y. Gao, and W. e. a. Wang, "A hybrid ant colony algorithm based on multiple strategies for the vehicle routing problem with time windows," *Complex Intell. Syst.*, vol. 9, pp. 2491–2508, 2023.

[6] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *Biosystems*, vol. 43, no. 2, pp. 73–81, 1997.

[7] L. M. Gambardella, E. Taillard, and G. Agazzi, *Ant Colonies for Vehicle Routing Problems*. London, UK: McGraw-Hill, 1999, pp. 63–76.

[8] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.

[9] J.-Y. Potvin and S. Bengio, "The vehicle routing problem with time windows part ii: Genetic search," *INFORMS Journal on Computing*, vol. 8, no. 2, pp. 165–172, 1996.

[10] K. C. Tan, L. H. Lee, Q.-L. Zhu, and K. Ou, "Heuristic methods for vehicle routing problem with time windows," *Artificial Intelligence in Engineering*, vol. 23, no. 1, pp. 309–315, 2005.

[11] T. Dokeroglu, A. Deniz, and H. Kiziloz, "A comprehensive survey on recent metaheuristics for feature selection," *Neurocomputing*, vol. 494, pp. 269–296, 2022.

[12] R. Azam, M. Riaz, M. Farooq, F. Ali, M. Mohsan, and A. D. et al., "Optimization-based economical flexural design of singly reinforced concrete beams: a parametric study," *Materials (Basel)*, vol. 15, p. 3223, 2022.

[13] A. Dalvand, M. Dowlatshahi, and A. Hashemi, "Sgfs: a semi-supervised graph-based feature selection algorithm based on the pagerank algorithm," in *27th International Computer Conference, Computer Society of Iran (CSICC)*, 2022, pp. 1–6.

[14] G. Bekdas, S. Nigdeli, A. Kayabekir, and X.-S. Yang, *Optimization in civil engineering and metaheuristic algorithms: a review of state-of-the-art developments*. Springer, 2022, pp. 111–137.

[15] C. C. Sartori, C. Blum, F. Bistaffa, and G. R. Corominas, "Metaheuristics and large language models join forces: Towards an integrated optimization approach," *arXiv preprint*, vol. 2405.18272, 2024, submitted for publication in an international journal. [Online]. Available: https://doi.org/10.48550/arXiv.2405.18272