ADVANCED START METHOD FOR SOLVING VEHICLE ROUTING PROBLEMS

ỨNG DỤNG PHƯƠNG PHÁP KHỞI ĐẦU NÂNG CAO NHẰM NÂNG CAO HIỆU QUẢ GIẢI QUYẾT BÀI TOÁN ĐỊNH TUYẾN

Nguyen Duc Duy*, Lao Khai Kien, Phung Gia Bao

Ho Chi Minh City University of Technology, VNU-HCM, Vietnam

*Corresponding author: duy.nguyen@hcmut.edu.vn

(Received: July 03, 2025; Revised: August 13, 2025; Accepted: August 20, 2025)

DOI: 10.31130/ud-jst.2025.23(9A).360E

Abstract - This study addresses a routing problem at a last-mile company in Vietnam. It proposes using a machine learning technique-Nearest Neighbor Search-to identify the most similar known solution, which is then used as the initial solution for a branch-and-bound algorithm. Firstly, the machine learning investigated a dataset of 500 pre-solved problems to find solutions for new problems. Then, the solution found is fed to the solver as a hint. The experimental results indicate that the algorithm can retrieve solutions for similar cases in just a few milliseconds. For entirely new problems, the method reduces computation time compared to the traditional branch-and-bound algorithm. This significant improvement in efficiency demonstrates the method's practical value in real-world logistics operations. The combination of machine learning and classical optimization in this approach shows promise for enhancing decision-making processes in the logistics industry, particularly in scenarios requiring rapid and reliable route planning.

Key words - Vehicle routing problem; branch-and-bound; machine learning; advanced start; nearest neighbor search

Tóm tắt - Nghiên cứu này tập trung vào giải quyết bài toán định tuyến tại một công ty giao chặng cuối ở Việt Nam. Phương pháp được đề xuất sử dụng một kỹ thuật trong học máy – Tìm kiếm lân cận gần nhất (Nearest Neighbor Search) – để xác định lời giải phù hợp nhất từ các bài toán tương tự đã biết, làm cơ sở tìm kiếm tối ưu trong giải thuật phân nhánh và chặn. Đầu tiên, học máy được sử dụng để phân tích và khai thác bộ dữ liệu gồm 500 bài toán đã giải làm cơ sở đối chiếu nhằm tìm kiếm lời giải gần nhất cho bài toán mới. Lời giải này được cung cấp cho phần mềm tối ưu như gợi ý ban đầu. Thực nghiệm cho thấy thuật toán có thể xác định lời giải cho bài toán tương tự chỉ trong vài mili giây. Đối với các bài toán mới, phương pháp đề xuất giúp tiết kiệm thời gian giải so với giải thuật phân nhánh và chặn truyền thống. Những kết quả này cho thấy thuật toán đề xuất có hiệu quả cao và tính ứng dụng tốt trong giải quyết các bài toán thực tiễn.

Từ khóa - Bài toán định tuyến; phân nhánh và chặn; học máy; khởi đầu nâng cao; tìm kiếm lân cận gần nhất

1. Introduction

The Capacitated Vehicle Routing Problem (CVRP) is one of the variants of the Vehicle Routing Problem (VRP), which is widely applied in last-mile delivery [1-3]. As shown in Figure 1, the CVRP shares the same concept as VRP while considering the capacity of the fleet for serving a set of customers, aiming to minimize the total route cost while ensuring that each customer's demand is met without exceeding vehicle capacities [4-7]. CVRP is beneficial for last-mile delivery since it improves operational costs and service quality [8-10]. Due to its importance, there is an investment in skilled professionals to design intelligent systems that can minimize both transportation and storage expenses, as well as computational resources [2, 11]. As a result, scholars have developed methods to address VRP and its variants, such as Simulated Annealing [4], Data-driven optimization [3], Local search [2], Integrated machine learning for optimization [12], etc.

This study presents a framework combining historical data with optimization techniques to improve the quality of routing solutions and contextual relevance. The proposed approach highlights the trade-off between computational time and solution quality, offering practical benefits in terms of both operational efficiency and cost reduction. Its effectiveness is demonstrated through experiments on a real-world dataset.

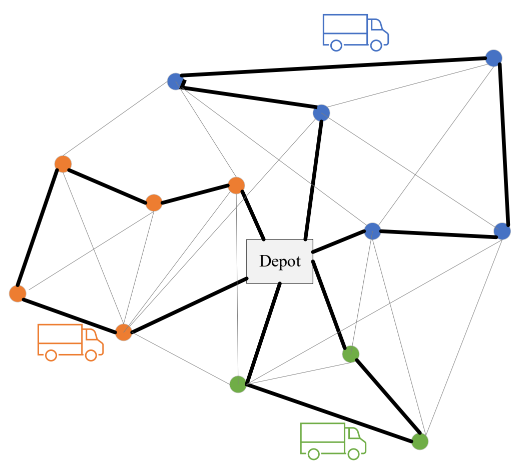

Figure 1. *Vehicle routing problem*

The key contributions of this study are as follows:

- A historical dataset of problem instances and corresponding solutions from a last-mile delivery company is constructed, serving as a reusable knowledge base for future routing tasks.

- A novel framework is proposed that incorporates a machine learning model and a Nearest Neighbor Search algorithm to retrieve relevant problem instances from the database based on similarity to a new problem.

- An advanced start strategy is employed to enhance the quality of initial solutions, which are then used to update and enrich the knowledge database.

2. Background

2.1. Capacitated Vehicle Routing Problem

The Vehicle Routing Problem (VRP), a well-known NP-hard problem, was first introduced by Dantzig and Ramser [13]. Solutions to the VRP and its variants are generally categorized into two types: exact and approximate, which can be seen in selective research.

Tiwari and Sharma [2] addressed vehicle routing problems with capacity constraints. This study compares heuristic algorithms-Intra-Route Local Search, Inter-Route Local Search, and Tabu Search-showing that while Tabu Search performs best on larger problem instances, local search methods are more efficient for smaller instances. İlhan [4] proposed an enhanced simulated annealing algorithm incorporating crossover operators (ISA-CO) for the CVRP problem. Tested on 91 benchmark instances, ISA-CO outperformed several state-of-the-art methods across many cases, demonstrating its effectiveness. Altabeeb *et al.* [14] addressed a CVRP by using a Cooperative Hybrid Firefly Algorithm (CVRP-CHFA) that incorporates multiple firefly populations. Experimental results on 108 benchmark instances demonstrate that CVRP-CHFA outperforms several existing hybrid firefly algorithms and delivers competitive performance compared to current methods. Chi *et al.* [15] investigated the 3D Loading Capacitated Vehicle Routing Problem under a relocation-ban constraint, proposing two improved relocation strategies to balance transportation cost and operational complexity. Using a mixed-integer linear programming model and an enhanced branch-and-price algorithm, the study demonstrates that allowing necessary relocations improves volume utilization by an average of 3.75% (up to 36.48%) and reduces total transportation cost by an average of 4.86% (up to 13.08%) in large-scale benchmark instances.

Luo and Li [16] presented a deep reinforcement learning framework combining a residual graph convolutional encoder and multiple attention-based decoders to solve CVRP. By using numerous decoder strategies and achieving solution quality, the framework was efficient in solving large-scale instances, the CVRPLIB benchmarks, and a real-world Jingdong Logistics distribution problem. Hao *et al.* [17] addressed capacitated vehicle routing problems with time windows (CVRPTW) on a road network. The objective is to minimize total operational costs while meeting delivery time constraints. A novel three-stage algorithm combining dynamic programming and a modified insertion heuristic efficiently solves the model. Numerical experiments demonstrate significant cost savings and improved energy efficiency from platooning, validating the practical benefits of the proposed approach. Liu *et al.* [18] proposed an Adaptive Genetic Algorithm with Elastic Strategy (AGA-ES) to effectively solve dynamic capacitated vehicle routing problems (DCVRPs), adapting to changes such as node increases, decreases, and road condition variations. Experimental results show AGA outperforms traditional methods on static CVRPs. At the same time, AGA-ES successfully handles dynamic demands, demonstrated through benchmark datasets and a real-world case with 100 SF express stations in Xi'an. Kyriakakis *et al.* [19] proposed a Hybrid Tabu Search - Variable Neighborhood Descent (HTS-VND) algorithm for the Cumulative Capacitated Vehicle Routing Problem with Time Windows (CCVRPTW) and its variant without time windows (CCVRP). Tested on 92 CCVRP instances, HTS-VND matched the best-known solutions in 84 cases and found new best solutions in two. For 56 CCVRPTW instances, HTS-VND outperformed two benchmark algorithms, achieving the specified number of vehicles in 41 instances and providing new solution cost data not previously reported.

2.2. Advanced Start for Vehicle Routing Problem

In combinatorial optimization problems like the VRP, the quality of the initial solution plays a critical role, especially in metaheuristic and hybrid approaches. An "advanced start" refers to generating a more informed or higher-quality initial solution to accelerate convergence and improve final solution quality. Unlike random or greedy starts, advanced starts incorporate domain knowledge, heuristic logic, or data-driven strategies [20, 21]. Some scholars have reported such as Alesiani et al. [22] proposed a Constrained Clustering Capacitated Vehicle Routing Solver (CC-CVRS). This method reduces problem complexity by transforming the CVRP into a soft-clustered version while preserving original constraints, enabling faster solutions with small optimality gaps. Experimental results show that CC-CVRS can solve large instances in seconds, making it suitable for near real-time applications or as a warm-start strategy for exact solvers. Alkaabneh *et al.* [23] addressed the inventory routing problem for perishable goods to maximize supplier profit while minimizing fuel, inventory holding costs, and greenhouse gas emissions. A Benders decomposition-based exact method is proposed, enhanced by acceleration techniques including valid inequalities and a GRASP-based warm-up strategy. Computational results on large-scale instances demonstrate both high solution quality and efficiency, with 2–11% fuel savings when using a detailed fuel consumption model compared to traditional distance-based models. Aguayo *et al.* [24] explored the Vehicle Routing Problem with Transfers (VRP-T), an extension of the Split Delivery VRP (SDVRP), where customer locations can also serve as transfer points for exchanging loads between vehicles. A two-index mixed-integer programming (MIP) model, supported by a heuristic warm-start approach, is developed to solve the problem. Computational results show that allowing transfers can reduce routing costs by up to 50% and is at least as effective as split deliveries in minimizing costs and vehicle usage. Najy *et al.* [25] investigated the integration of collaborative truck–drone delivery into the Inventory Routing Problem (IRP), proposing a novel IRP with Drones (IRP-D) model. A mixed-integer linear programming formulation and a branch-and-cut exact solution method are developed and supported by a heuristic based on the classical IRP. Computational results demonstrate that the proposed heuristic performs effectively both independently and as a warm-start for the exact solver, highlighting the operational benefits of incorporating UAVs into inventory routing. Fragkogios *et al.* [26] addressed the Multi-Trip Time-Dependent Vehicle Routing Problem with Time Windows (MTTDVRPTW) by proposing a reduced-size reformulation solved through an accelerated Benders decomposition algorithm. The method incorporates novel valid inequalities and a warm start strategy to tighten bounds and reduce infeasibility in the master problem. Computational results demonstrate that the proposed approach outperforms non-decomposed models and existing heuristics in both efficiency and solution quality.

Despite the progress, several gaps remain. First, there is a lack of generalized frameworks for generating advanced starts across VRP variants. Second, data-driven methods remain underutilized, especially for large-scale or dynamic VRPs. Lastly, many advanced start strategies are still designed manually, suggesting potential for more automated or adaptive systems in future research.

3. Methodology

3.1. Mathematical model

This section presents a typical CVRP that is used in this research [27]. There is a set of customers V = {1, 2,..., n}, and a depot node 0. The nodes generate a complete graph G = (V ∪ {0}, E).

Parameter:

- d_{ij}: cost (or distance) from node i to node j;

- c: cost per km;

- Tr: cost per truck;

- D_i: demand of customer I;

- Q: capacity of each vehicle;

- K: number of vehicles (optional, assume enough vehicles).

Decision Variables:

- $x_{ij} \in \{0,1\}$: 1 if the arc from node i to node j is used, 0 otherwise;

- u_i: Load or position the vehicle in the route upon arriving at customer I;

- To manage the complexity of the study, the following simplifying assumptions have been made;

- Each customer's order must be delivered by a single vehicle; no splitting of deliveries is allowed;

- Once delivery routes are assigned, they remain unchanged; any new orders will be scheduled in future delivery cycles.

The model does not account for pickups, product returns, or customer complaints.

- Variations in load size, vehicle capacity, and warehouse constraints are not considered;

- Backhauling activities are excluded from the analysis;

- The study assumes a deterministic setting, with no random or uncertain factors involved.

Objective

Minimize:

$$c\times\sum_{i=0}^{i=n}\sum_{j=0}^{j=n} d_{ij}\times x_{ij} + Tr\times\sum_{j=1}^{j=n} x_{0j}$$

Subject to:

Each customer is visited exactly once:

$$\sum_{j=0}^{j=n} x_{ij}=1 \qquad \forall\ i = 1,...,n \tag{1}$$

$$\sum_{i=0}^{i=n} x_{ij}=1 \qquad \forall\ j = 1,...,n \tag{2}$$

Vehicle flow conservation at the depot:

$$\sum_{j=1}^{j=n} x_{0j}\leq K \tag{3}$$

$$\sum_{i=1}^{i=n} x_{i0}\leq K \tag{4}$$

Subtour elimination (MTZ constraints):

$$u_i - u_j + Q \times x_{ij} \leq Q - D_j \qquad \forall\ i \neq j,\ i, j \in \{1,...,n \tag{5}$$

$$D_i \leq u_i \leq Q \qquad \forall\ i \in \{1,...,n\} \tag{6}$$

Binary constraints:

$$x_{ij} \in \{0,1\} \qquad \forall i, j \in \{0,...,n\} \tag{7}$$

3.2. Nearest Neighbor Search

Nearest Neighbor Search (NNS) is one of the fundamental techniques in machine learning used to identify data points in a high-dimensional space closest to a given query point. The main idea is to measure the similarity or distance between data points using metrics such as Euclidean distance or cosine similarity [28].

Let $X = \{x_1, x_2,..., x_n\} \subset \mathbb{R}^d$ be a dataset of n points in a d-dimensional Euclidean space, and let q be a query point in $\mathbb{R}^d$.

The goal of Nearest Neighbor Search (NNS) is to find:

$$x^*=argmin_{(x_i \in X)} dist\ (q, x_i)$$

Where $dist$: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}$ is a distance metric. Common choices include:

Euclidean distance:

$$dist(q, x_i)=\|q-x_i\|_2=\left[\sum_{j=1}^{d}\left(q_j - x_{ij}\right)^2\right]^{1/2}$$

Cosine distance:

$$dist(q, x_i)=1-\frac{\langle q,x_i\rangle}{\|q\|_2\|x_i\|_2}$$

Manhattan distance:

$$dist(q, x_i)=\|q-x_i\|_1=\sum_{j=1}^{d} |q_j-x_{ij}|$$

3.3. Advanced start methodology

When solving Mixed Integer Programming (MIP) problems, optimization solvers such as Gurobi or IBM CPLEX can leverage user-provided initial solutions-commonly referred to as MIP starts, warm starts, or advanced starts. These inputs consist of partial or complete sets of variable assignments representing candidate solutions, which may originate from prior problem instances or domain-specific heuristics. While MIP starts may correspond to feasible solutions, feasibility is not a strict requirement; solvers can also utilize incomplete or infeasible MIP starts to guide the initial stages of the search process and potentially accelerate convergence.

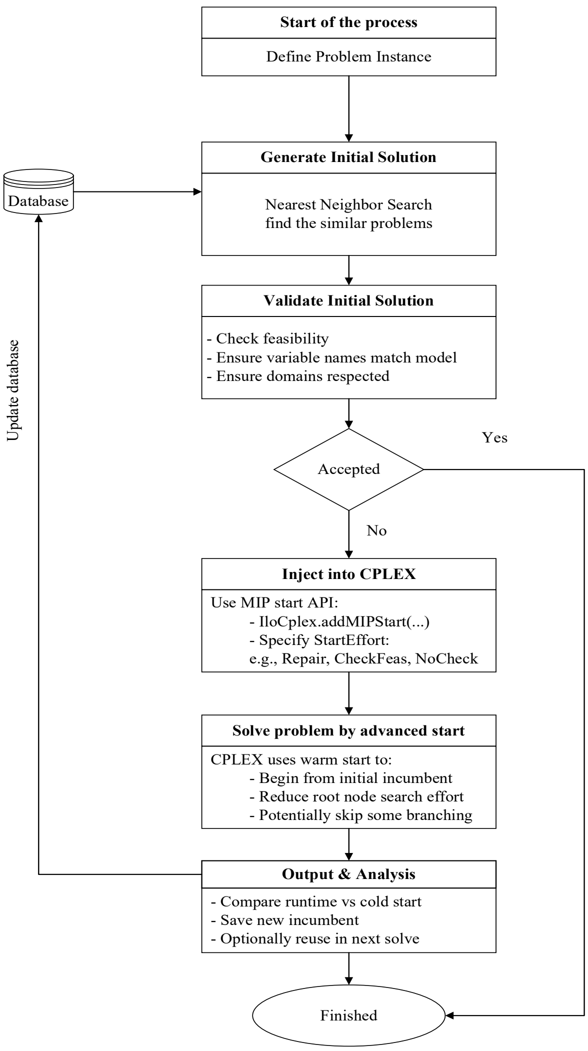

Figure 2. *Advanced start methodology*

Figure 2 illustrates a workflow that integrates Nearest Neighbor Search (NNS) as a mechanism for knowledge reuse in solving MIP problems. The process begins with the arrival of a new problem instance, typically characterized by a set of customers and their associated demands. Subsequently, an initial feasible solution is retrieved from a repository of previously solved, structurally similar problems. This candidate solution undergoes a rigorous validation stage to confirm its feasibility, ensure full compliance with the variable naming conventions of the model, and verify adherence to the prescribed variable domains and integrity conditions. Solutions that fail validation are either refined or replaced through iterative regeneration, with the database being updated accordingly. Once validated, the solution is checked by the user; if they are accepted, this solution will become the solution for the new problem; otherwise, the solution is injected into CPLEX using the MIP start interface (e.g., IloCplex.addMIPStart(...)), accompanied by an appropriate StartEffort parameter setting (e.g., Repair, CheckFeas, or NoCheck), which governs the extent of feasibility checks and repair operations performed by the solver. CPLEX then initiates the branch-and-bound search from the provided incumbent solution, thereby reducing root node processing time, decreasing the number of explored nodes, and potentially bypassing certain branching decisions. The final stage involves performance analysis, wherein solution quality and runtime are compared against baseline cold-start runs. Improved incumbents are archived for potential reuse in subsequent problem instances, thereby creating a feedback loop that continuously enhances the solver's starting point for related optimization tasks.

4. Computational results

The Vehicle Routing Problem (VRP) model was implemented using the IBM Decision Optimization CPLEX Modeling for Python (DOcplex) library. The Nearest Neighbor Search (NNS) algorithm was developed and executed using Python within a Jupyter Notebook environment. All computational experiments were conducted on a machine equipped with an Intel Core i7 155H processor, 32 GB of RAM, and running the Windows 11 operating system.

4.1. Data

Table 1. *Customer information*

Customer ID	Latitude	Longitude	Demand (kg)
1	10.85769	106.7412	60
2	10.87105	106.7092	150
3	10.92364	106.77	150
4	10.9671	106.734	90
5	10.8634	106.7616	30
…	…	…	…
19	10.8803	106.7266	10
20	10.84095	106.7972	20

This study investigates a transportation network belonging to a logistics company in Ho Chi Minh City, Vietnam. The company offers warehousing and distribution services to 20 customers across the city, as shown in Table 1. The distance matrix between the depot (located at 10.8230989, 106.6296638) and customer locations is computed using their geographic coordinates (latitude and longitude) obtained from Google Maps, as shown in Figure 3. The company operates 5 trucks, each with a 720 kg capacity, capable of making deliveries throughout the city during both peak and off-peak hours. Each day, the planning team at the warehouse processes customer orders and assigns delivery routes to the trucks. Drivers then load the products, organized per customer using the Last-In-First-Out (LIFO) principle, and deliver. According to the manufacturer's specifications, the operating cost is 8,000 VND/km. The fixed cost of using a truck is 500,000 VND.

	Depot	1	2	3	4	5	…	19	20
Depot	0.0	9.3	6.7	15.5	11.1	11.6	…	8.8	10.9
1	9.8	0.0	5.8	9.9	14.1	6.3	…	5.0	8.4
2	7.2	5.8	0.0	10.8	13.0	7.8	…	4.2	12.2
3	12.0	9.9	10.8	0.0	8.2	8.7	…	8.7	11.6
4	9.6	14.1	13.0	8.2	0.0	11.9	…	11.6	17.6
5	10.1	6.3	7.8	8.7	11.9	0.0	…	6.3	6.6
…	…	…	…	…	…	…	…	…	…
19	9.3	5.0	4.2	8.7	11.6	6.3	…	0.0	10.9
20	11.4	8.4	12.2	11.6	17.6	6.6	…	10.9	0.0

Figure 3. *Distance matrix (km)*

4.2. Result

This section illustrates the principle of Nearest Neighbor Search through a simple example to identify the most similar problem instance in a database. The database is represented by a data matrix, where each row corresponds to a problem instance and includes attributes such as total demand, the number of vehicles used, and so on. A new problem instance, denoted as vector A, contains the same type of information. The objective is to compare vector A with the existing entries in the data matrix to identify the closest match based on a similarity metric.

A=[Total demand, number of used vehicle, customer demand]

To perform this task, the KNeighborsClassifier from the *scikit-learn* library in Python is employed with k = 1, and the distance metric is Euclidean distance, effectively retrieving the single most similar case.

To illustrate the methodology, an application example of the advanced start methodology is presented in Table 2. Then the route turns to mathematical solutions. The following notation is used: *C* is the list of customers, D is customer demand, B is the routing set, and R is the solution obtained after the *Retrieval* step. Assuming that the capacities of the vehicles are respected and equal to 720 kg.

To prepare the database, a number of problems are generated using random demand values D ~ U (10, 150). Each instance is solved to obtain its optimal solution. Besides, the author also utilizes available problems and solutions that are used in the company. All is then stored in the database to support a case-based reasoning approach. Such a database consists of 500 problems. We have a test with 20 problems, and our method can define a solution for almost all new problems.

In this work, an initial problem instance retrieved from the database has the following characteristics:

- ***Problem 1-Advanced start 1*****:** Total demand of 1930 kg, number of required vehicles equal to 3, mean demand of 96.5 kg, minimum demand of 30 kg, and maximum demand of 150 kg.

- ***Problem 2- Advanced start 2:*** Total demand of 1410 kg, number of required vehicles equal to 2, mean demand of 70.5 kg, minimum demand of 10 kg, and maximum demand of 130 kg.

A new problem instance to be solved shares similar statistical features-mean demand of 56.5 kg, minimum of 10 kg, and maximum of 120 kg-but with a total demand of 1130 kg and only 2 required vehicles. CPLEX also solves it without an initial solution – Coldstart.

Table 2. *Sample case used to illustrate the methodology*

Initial data	
C =	$\{c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$
D =	[0 150 150 140 120 90 100 60]
Retrieval	
C =	$\{c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$
D =	[0 140 130 100 90 80 100 100]
R =	r_1: $[\mathbf{c_0} \rightarrow c_2 \rightarrow c_4 \rightarrow c_5 \rightarrow c_7 \rightarrow \mathbf{c_0}]$,
	r_2: $[\mathbf{c_0} \rightarrow c_1 \rightarrow c_3 \rightarrow c_6 \rightarrow \mathbf{c_0}]$

Table 3. *Advanced starts and Coldstart for VRP*

Time (secs)	Objective (VND)		
	Advanced start 1	Advanced start 2	Cold start
0	2,388,000	1,816,000	2,792,000
1	1,888,000	1,816,000	1,848,000
5	1,872,000	1,816,000	1,824,000
10	1,864,000	1,816,000	1,824,000
100	1,864,000	1,816,000	1,824,000
500	1,824,000	1,816,000	1,816,000
1000	1,824,000	1,816,000	1,816,000
1500	1,824,000	1,816,000	1,816,000
2000	1,824,000	1,816,000	1,816,000
3000	1,824,000	1,816,000	1,816,000
3600	1,824,000	1,816,000	1,816,000

Table 3 compares three solution approaches-Advanced Start 1 and 2, and Coldstart (running without an initial solution)-based on their objective values (in VND) over time. Among the three, Advanced Start 2 demonstrates the best performance, immediately achieving the optimal objective value of 1,816,000 VND at time 0 and maintaining it throughout the 3600-second interval. In contrast, Advanced Start 1 starts from a higher value of 2,388,000 VND but shows rapid improvement, reaching the optimal value by 500 seconds. Meanwhile, Coldstart begins with the worst initial solution (2,792,000 VND) and requires the same amount of time, about 500 seconds, to reach the optimal value, making it the slowest in convergence. Overall, providing a good initial solution makes it the most efficient method for applications requiring quick, high-quality solutions.

Figure 3 demonstrates the benefits of using an advanced start approach to optimize a Vehicle Routing Problem (VRP) in the first 100 out of 3600 seconds. By leveraging the similarity between these instances, the advanced start method enables a more efficient and potentially faster solution process for the new problem.

Even though advanced start methods effectively improve the running time for optimization problems [29-31], however, this method is highly problem-dependent [32-36]. Because decision makers often use solutions from similar problems as starting points, the solver can quickly find an initial feasible solution. However, it may then take a long time to improve this solution, or fail to do so, for several reasons: the current solution might be only locally optimal while better solutions exist elsewhere; it might already be the best possible, but the solver needs time to confirm this; or there may be room for improvement, but the time limit for each iteration is too tight to allow further progress.

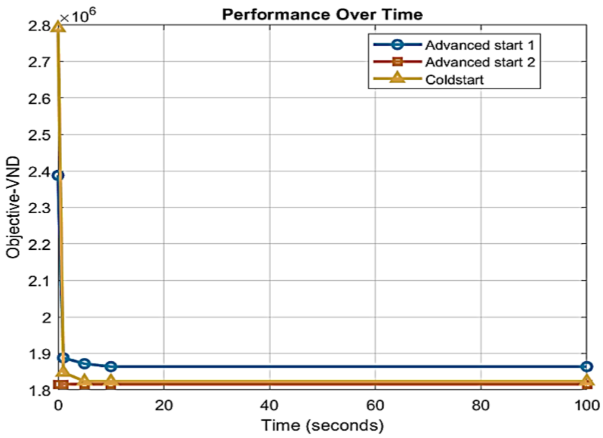

Figure 3. *Benefits of an advanced start*

5. Conclusion and future direction

This paper introduced an advanced start framework for solving a vehicle routing problem using data from a real case study in Vietnam. The problem includes many practical attributes and constraints, making it difficult to solve directly with a standard MIP solver, which often requires long computation times and yields solutions of low quality. To address this, we proposed a framework that combines machine learning and case-based reasoning to retrieve the most similar historical problems from a database, generate high-quality initial solutions, and then improve them by warm-starting the MIP solver. We demonstrated this approach on four real-world instances, and the results show that the proposed model outperforms both the direct MIP approach and the use of the machine learning model alone.

One limitation of the current method for future work is that it does not yet handle problems of varying sizes, such as differing numbers of customers. Therefore, future research will focus on extending the framework to handle problem instances with greater diversity in size and complexity. In addition, this study used only a single MIP solver with default parameter settings. Since various commercial MIP solvers are faced with difficulties when dealing with larger CVRP instances (e.g., 50+ customers), another promising direction is to explore the performance of metaheuristics, such as genetic algorithms, simulated annealing, or tabu search, with customized parameter settings, further to enhance solution quality and efficiency for different problem instances.

Acknowledgement: This research is funded by Ho Chi Minh City University of Technology – VNU-HCM, under grant number To-CK-2024-01. We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, for supporting this study.

REFERENCES

[1] C. K. Heng, A. N. Zhang, P. S. Tan, and Y.-S. Ong, "Multi-objective heterogeneous capacitated vehicle routing problem with time windows and simultaneous pickup and delivery for urban last mile logistics", in *Proc. the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, vol. 1, 2015: Springer, pp. 129-140.

[2] K. V. Tiwari and S. K. Sharma, "An optimization model for vehicle routing problem in last-mile delivery", *Expert Systems with Applications*, vol. 222, p. 119789, 2023.

[3] Z. Nourmohammadi, B. Hu, D. Rey, and M. Saberi, "A data-driven preference learning approach for multi-objective vehicle routing problems in last-mile delivery", *Transportation Research Part C: Emerging Technologies*, vol. 174, p. 105101, 2025.

[4] İ. İlhan, "An improved simulated annealing algorithm with crossover operator for capacitated vehicle routing problem", *Swarm and Evolutionary Computation*, vol. 64, p. 100911, 2021.

[5] C. E. Gounaris, W. Wiesemann, and C. A. Floudas, "The robust capacitated vehicle routing problem under demand uncertainty", *Operations research*, vol. 61, no. 3, pp. 677-693, 2013.

[6] Z. Borcinova, "Two models of the capacitated vehicle routing problem", *Croatian Operational Research Review*, vol. 8, pp. 463-469, 2017.

[7] T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter, "On the capacitated vehicle routing problem", *Mathematical programming*, vol. 94, pp. 343-359, 2003.

[8] R. I. Muslem and M. K. Nasution, "Algorithms and Approaches for the Vehicle Routing Problem with Pickup and Delivery (VRPPD): A Survey", in *Proc. 2024 Ninth International Conference on Informatics and Computing (ICIC)*, 2024: IEEE, pp. 1-5.

[9] G. D. Konstantakopoulos, S. P. Gayialis, and E. P. Kechagias, "Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification", *Operational research*, vol. 22, no. 3, pp. 2033-2062, 2022.

[10] F. Yang and F. Tao, "A bi-objective optimization VRP model for cold chain logistics: Enhancing cost efficiency and customer satisfaction", *IEEE Access*, vol. 11, pp. 127043-127056, 2023.

[11] S. Pan, V. Giannikas, Y. Han, E. Grover-Silva, and B. Qiao, "Using customer-related data to enhance e-grocery home delivery", *Industrial Management & Data Systems*, vol. 117, no. 9, pp. 1917-1933, 2017.

[12] J. Fitzpatrick, D. Ajwani, and P. Carroll, "A scalable learning approach for the capacitated vehicle routing problem", *Computers & Operations Research*, vol. 171, p. 106787, 2024.

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

[14] A. M. Altabeeb, A. M. Mohsen, L. Abualigah, and A. Ghallab, "Solving capacitated vehicle routing problem using cooperative firefly algorithm", *Applied Soft Computing*, vol. 108, p. 107403, 2021.

[15] J. Chi, S. He, and R. Song, "Solving capacitated vehicle routing problem with three-dimensional loading and relocation constraints", *Computers & Operations Research*, vol. 173, p. 106864, 2025.

[16] J. Luo and C. Li, "An efficient encoder-decoder network for the capacitated vehicle routing problem", *Expert Systems with Applications*, vol. 278, p. 127311, 2025.

[17] Y. Hao, Z. Chen, X. Sun, and L. Tong, "Planning of truck platooning for road-network capacitated vehicle routing problem", *Transportation Research Part E: Logistics and Transportation Review*, vol. 194, p. 103898, 2025.

[18] J. Li, R. Liu, and R. Wang, "Handling dynamic capacitated vehicle routing problems based on adaptive genetic algorithm with elastic strategy", *Swarm and Evolutionary Computation*, vol. 86, p. 101529, 2024.

[19] N. A. Kyriakakis, I. Sevastopoulos, M. Marinaki, and Y. Marinakis, "A hybrid Tabu search–Variable neighborhood descent algorithm for the cumulative capacitated vehicle routing problem with time windows in humanitarian applications", *Computers & Industrial Engineering*, vol. 164, p. 107868, 2022.

[20] M. Karimi-Mamaghan, M. Mohammadi, P. Meyer, A. M. Karimi-Mamaghan, and E.-G. Talbi, "Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art", *European Journal of Operational Research*, vol. 296, no. 2, pp. 393-422, 2022.

[21] S. Voigt, "A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems", *European Journal of Operational Research*, vol. 322, no. 2, pp. 357-375, 2025.

[22] F. Alesiani, G. Ermis, and K. Gkiotsalitis, "Constrained clustering for the capacitated vehicle routing problem (CC-CVRP)", *Applied artificial intelligence*, vol. 36, no. 1, p. 1995658, 2022.

[23] F. Alkaabneh, A. Diabat, and H. O. Gao, "Benders decomposition for the inventory vehicle routing problem with perishable products and environmental costs", *Computers & Operations Research*, vol. 113, p. 104751, 2020.

[24] M. M. Aguayo, F. N. Avilés, S. C. Sarin, and C. Archetti, "The vehicle routing problem with transfers", *Computers & Operations Research*, vol. 177, p. 106980, 2025.

[25] W. Najy, C. Archetti, and A. Diabat, "Collaborative truck-and-drone delivery for inventory-routing problems", *Transportation Research Part C: Emerging Technologies*, vol. 146, p. 103791, 2023.

[26] A. Fragkogios, Y. Qiu, G. K. Saharidis, and P. M. Pardalos, "An accelerated benders decomposition algorithm for the solution of the multi-trip time-dependent vehicle routing problem with time windows", *European Journal of Operational Research*, vol. 317, no. 2, pp. 500-514, 2024.

[27] S. Irnich, P. Toth, and D. Vigo, "Chapter 1: The family of vehicle routing problems", in *Vehicle Routing: Problems, Methods, and Applications*, Second Edition: SIAM, 2014, pp. 1-33.

[28] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality", in *Proc. the thirtieth annual ACM symposium on Theory of computing*, 1998, pp. 604-613.

[29] E. B. Edis and I. Ozkarahan, "A combined integer/constraint programming approach to a resource-constrained parallel machine scheduling problem with machine eligibility restrictions", *Engineering Optimization*, vol. 43, no. 2, pp. 135-157, 2011.

[30] S. M. Pour, J. H. Drake, L. S. Ejlertsen, K. M. Rasmussen, and E. K. Burke, "A hybrid constraint programming/mixed integer programming framework for the preventive signaling maintenance crew scheduling problem", *European Journal of Operational Research*, vol. 269, no. 1, pp. 341-352, 2018.

[31] G.-H. Chen, J.-C. Jong, and A. F.-W. Han, "Applying constraint programming and integer programming to solve the crew scheduling problem for railroad systems: Model formulation and a case study", *Transportation Research Record*, vol. 2676, no. 1, pp. 408-420, 2022.

[32] C. Blum, "Construct, merge, solve and adapt: Application to the minimum global domination problem", *TOP*, pp. 1-21, 2024.

[33] F. Ornelas, A. Santiago, J. A. Castan Rocha, S. Ibarra Martínez, and A. H. García, "Warm Starting Integer Programming for the Internet SHopping Optimization Problem with Multiple Item Units (ISHOP-U)", in *Artificial Intelligence in Prescriptive Analytics: Innovations in Decision Analysis, Intelligent Optimization, and Data-Driven Decisions*, 2024, pp. 153-170.

[34] A. Etminaniesfahani, H. Gu, L. M. Naeni, and A. Salehipour, "An efficient relax-and-solve method for the multi-mode resource constrained project scheduling problem", *Annals of Operations Research*, vol. 338, no. 1, pp. 41-68, 2024.

[35] W. Murray and H. Shek, "A local relaxation method for the cardinality constrained portfolio optimization problem", *Computational Optimization and Applications*, vol. 53, pp. 681-709, 2012.

[36] E. John and E. A. Yıldırım, "Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension", *Computational Optimization and Applications*, vol. 41, no. 2, pp. 151-183, 2008.