**Implementation of Sustainable Technologies in Transportation Planning for Agricultural Logistics Using a Hybrid Evolutionary Algorithm**

Larisa Sulyukova1, a), Zarrina Akhmedjanova2, b), Rasul Shamsiev3, c)

1***Tashkent Institute of Irrigation and Agricultural Mechanization Engineers – National Research University, Tashkent 100000, Uzbekistan***2*Research Institute for the Development of Digital Technologies and Artificial Intelligence, 17A, Buz-2, Tashkent, 100125, Uzbekistan*3*Tashkent State Transport University, 1 Temiryulchilar St., Tashkent 100167, Uzbekistan*

*a) Corresponding author: [l.sulyukova@tiiame.uz](mailto:l.sulyukova@tiiame.uz)  
b)[zarrina92@inbox.ru](mailto:zarrina92@inbox.ru)  
c)rasulshamsiev@rambler.ru*

**Abstract.** This paper presents a research-oriented approach to optimizing transportation routing in integrated agricultural logistics using a hybrid evolutionary algorithm. The study addresses constraints typical of the agricultural sector, such as delivery deadlines, vehicle capacity, and strict loading/unloading order requirements. A hybrid algorithm is proposed, combining Differential Evolution, a Genetic A hybrid algorithm is proposed, combining Differential Evolution and a Genetic Algorithm with Local Search centered on Variable Neighborhood Search (VNS). A mathematical model and cost-based objective function are introduced, accounting for transportation, time, and operational constraints. To ensure convergence, a termination criterion based on the relative change in solution cost is applied. Numerical experiments on instances of up to 200 requests show an 8–12% cost reduction compared to baseline heuristics, with consistent solution stability. The reduction in mileage and number of routes contributes to lower fuel consumption and emissions, supporting the environmental sustainability of the approach. Additionally, two alternative local search methods ‒ a Gradient-based (Steepest Descent) Local Search and a discrete 2-opt algorithm ‒ are evaluated. Their integration further enhances the performance and robustness of the hybrid search. The results confirm the effectiveness of the proposed hybrid evolutionary method in addressing complex agricultural transportation logistics problems.

**Keywords:** transportation logistics, agricultural sector, hybrid algorithm, differential evolution, LIFO, VNS, 2-opt

# **INTRODUCTION**

The shift toward smart logistics in agriculture requires integrating heterogeneous data ‒such as product batches, vehicle conditions, schedules, and regulatory constraints ‒ into transportation systems. Traditional exact optimization methods (e.g., mixed-integer or dynamic programming) are often infeasible due to high dimensionality and data uncertainty. While evolutionary algorithms are effective for NP-hard routing problems, basic Differential Evolution (DE) may lack solution diversity and fail to escape local optima without hybridization [1]. This is particularly problematic in structurally constrained problems, such as Vehicle Routing Problems with Backhauls (VRPB). Research indicates that combining Genetic Algorithms (GA) with local search greatly improves solution quality [2], yet transport logistics systems still lack integrated approaches that jointly address data and route optimization.

This study addresses transportation route planning in the agricultural sector, which presents unique challenges such as time-sensitive deliveries, heterogeneous vehicle fleets, and strict loading and delivery constraints [2]. These features require adapted models and algorithms [2, 3]. To meet the need for data-driven decision support and advanced optimization under multiple constraints, we propose a hybrid evolutionary approach. It combines Differential Evolution, Genetic Algorithms, and local search techniques. The subsequent sections present the problem formulation, the mathematical model, the developed algorithm, and the experimental results, demonstrating enhanced efficiency and environmental sustainability in transport planning.

# **FORMULATION OF THE PROBLEM AND MATHEMATICAL MODEL**

**Problem Statement**

The problem addressed in this work is categorized as a Capacitated Vehicle Routing Problem with Time Windows and Last-In-First-Out (LIFO) loading/unloading constraints (CVRPTW-LIFO). This represents an extension of the classical Capacitated Vehicle Routing Problem (CVRP), enhanced with realistic constraints typical for agricultural logistics. In the standard CVRP formulation, a fleet of vehicles must fulfill a set of delivery requests without exceeding the capacity of each vehicle, typically limited by weight or volume [2]. The extended version ‒Vehicle Routing Problem with Time Windows (VRPTW) - adds temporal constraints, requiring that each delivery be made within a specified time interval or by a strict deadline [2, 3].

In our case, an additional constraint is introduced on the loading and unloading order. LIFO (Last-In, First-Out) rules are common in agricultural logistics to prevent cargo rearrangement or damage—goods loaded last must be unloaded first. This significantly narrows the set of feasible routes, as deliveries must follow the reverse order of loading. These constraints, categorized as Vehicle Routing Problems with Loading Constraints (VRP-LC), are well-studied and typically require specialized heuristics or metaheuristics for effective resolution [4, 5].

Based on recent research and VRP classifications, our problem is categorized as a Capacitated Vehicle Routing Problem with Time Windows and LIFO Loading Constraints (CVRPTW-LIFO) [1, 3]. It incorporates three types of complex constraints: Capacity constraints, limiting the weight and volume that each vehicle can carry; Time window constraints, requiring that each delivery be completed within a specified time interval or by a given deadline, with delays recurring penalties or infeasibility; LIFO loading and unloading constraints [6, 10], which mandate that deliveries occur in the exact reverse order of loading.

The combination of these constraints makes the problem NP-hard and difficult to solve exactly, which justifies the use of advanced heuristics—particularly hybrid evolutionary algorithms and local search techniques [6]. Further complexity stems from the need to integrate and synchronize data across information system modules (e.g., warehouse, transport, orders), a crucial requirement in agricultural logistics, where coordination between harvesting and delivery is time-sensitive [6, 7]. Given these challenges, hybrid evolutionary strategies ‒combining Differential Evolution (DE), Genetic Algorithms (GA), and local search ‒ represent a practical and effective solution for planning and managing agricultural transportation [1, 2, 3, 4, 8].

**Mathematical Model**

We now outline the mathematical optimization model for the described problem. The transportation network is represented as a directed graph , where is the set of nodes (pickup/delivery locations and the depot), and is the set of roads with specified distances or travel times between nodes *i* and *j*. A set of delivery requests is given, where each request is defined by its origin and destination (in the simplest case, from a central warehouse, i.e., only a destination). For each request *i*, the weight, volume , and delivery deadline (or time window) are specified. There is a fleet of vehicles located at the depot. Each vehicle *V* has a limited load capacity (by weight), volume capacity , a travel speed (used to compute travel time), and a fixed usage cost (e.g., cost per trip).

The objective is to determine a set of routes for a selected subset of vehicles, where each route is an ordered sequence of requests assigned to a specific vehicle .

The objective of optimization is to minimize the total logistics cost, including transportation expenses, route activation costs, and penalties for violations of delivery deadlines or loading/unloading order. Accordingly, an integrated objective function , is minimized, encompassing: transportation costs (i.e., distance-based or fuel consumption–based expenses for each route); fixed costs for vehicle activation (e.g., driver wages, vehicle depreciation); penalties for missing delivery deadlines, violations of the LIFO rule and overloads in weight or volume. The resulting objective function is formulated as:

(1)

where denotes the set of vehicles, represents the ordered set of routes, is a binary variable indicating whether vehicle is assigned to route , is the cost of transporting goods by vehicle along route , is the fixed cost of using the route, is the delay penalty function, is the weight coefficient for deadline adherence (with applied to perishable goods), is the penalty weight for technological constraints, denotes the cumulative penalty for violations of LIFO rules, weight or volume limits, and other operational constraints.

Subject to the following constraints:

- capacity and load weight limitations:

(2)

where is a binary variable that indicates whether cargo is assigned to vehicle and denote the weight and volume of cargo, respectively; is the set of available vehicles; is the load capacity (by weight) of vehicle ; is the volumetric capacity of vehicle (in volume);

- delivery deadline constraints:

(3)

where is the time cargo remains in vehicle ; is the delivery deadline; is the total duration of the route; is the set of all delivery requests.

**-** LIFO (Last-In, First-Out) loading/unloading constraints:

If , then cargo must be unloaded after cargo , assuming both are assigned to the same vehicle:

(4)

In addition, each cargo item must be delivered by one and only one vehicle: **,** that may operate no more than one route within the planning period, i.е. .

# **HYBRID EVOLUTIONARY ALGORITHM**

To solve the above-defined problem, we developed a hybrid evolutionary algorithm that integrates Differential Evolution [11], a Genetic Algorithm, and a Local Search centered on Variable Neighborhood Search (VNS) [5] local optimizer (denoted DE–GA+LS). The choice of this combination is motivated by the exponential complexity of exact methods (brute force, integer programming) which renders them impractical for large-scale instances. The proposed hybrid scheme balances a global evolutionary search with intensive local refinement, yielding near-optimal solutions in reasonable time for the problem sizes of interest.

Each candidate solution (individual *π*) is encoded as a route chromosome, representing routes for all vehicles:

(5)

We use a permutation-based encoding with route delimiters. A chromosome is essentially a sequence of order identifiers separated by special 0 markers that denote the depot and split the sequence into separate vehicle routes. A chromosome like [0, 5, 2, 8, 0, 3, 1, 0] encodes two routes, with 0 marking the depot. Each order appears once, allowing DE and GA operators to work efficiently on the permutation. For each solution, the total logistics cost *F* is calculated, including transportation expenses, route utilization costs, and penalties for constraint violations (time windows, capacity, and LIFO loading order).

The lateness penalty is defined as:

(6)

where is the actual arrival time at client *i*; if the delivery is on time, the penalty is zero.  
 denotes the client’s deadline — i.e., if , then no penalty is applied.

The penalty for violations of capacity constraints and the LIFO loading/unloading principle is calculated as:

(7)

where - total penalty function (penalty for overloads and technological violations on the route of vehicle *j*); - penalty coefficients for exceeding weight capacity, volume capacity, and for violating technological constraints, respectively; - weight of delivery order *i*; - volume of delivery order *i*; - weight capacity limit of vehicle *j*; - volume capacity limit of vehicle *j*; - positive part of *x*, i.e., *x* if *x* >0 , and 0 otherwise; - binary indicator of a technological constraint violation (e.g., LIFO rule) for vehicle *j* and order pair or constraint *k*. If the "last-in–first-out" rule is violated, the penalty coefficients corresponding to violations measured in kilograms, cubic meters, and monetary units, respectively.

The fitness (cost) of each individual solution is evaluated by computing the total cost of its routes, according to the objective function described in the mathematical model [12, 13]. This involves summing travel costs, usage costs, and any penalties for constraint violations. If a solution violates any hard constraints (capacity, time window, or LIFO order), it is considered infeasible; in such cases we assign a very large fitness value (by adding a large penalty) to effectively eliminate that individual from selection in the evolutionary process. Thus, only feasible or near-feasible solutions with minimal penalties tend to survive.

The DE–GA+LS algorithm is a population-based iterative process. Each generation of the algorithm applies a series of operators to evolve the population of route solutions. The main phases in each generation are [7, 9, 14]:

*Step 1: DE Mutation.*  For each target vector , a mutant vector is produced:

where (mutation factor) is self-adapted through Self-adaptive Differential Evolution (SaDE) mechanism.

*Step 2: DE Crossover*. A binomial DE crossover is performed between the target individual and the mutant vector, generating a trial individual

where is the j-th component of the new (trial) vector , generated based on the current vector and the mutant vector ; is the *j*-th component of the mutant vector (i.e., result of DE mutation); the *j*-th component of the target vector (the current individual in the population); is a random number uniformly distributed on the interval , independently generated for each component *j*; *CR* (Crossover Rate) is the recombination probability ‒ a predefined parameter that determines the probability of replacing component with .

*Step 3: GA Route Crossover (Elite Recombination).* Applied to the best individuals of the current population with a probability to recombine successful subroutes.

*Step 4: GA Mutation.* Introduces small random modifications with a mutation probability .

*Step 5:* *Local Search (LS).* After the global DE and GA operators, each candidate solution is further refined by an intensive local search procedure, which attempts to improve the solution by exploring its neighborhood in the solution space. In the constrained vehicle routing context, we incorporate multiple local search strategies to handle the complex constraints (capacity, time windows, and LIFO loading order) effectively:

**-** Variable Neighborhood Search (VNS) using successive neighborhoods (2-opt → relocate → swap).

2-opt removes intersecting edges in a route and reconnects segments to reduce overlap and distance. Relocate shifts an order to a new position, resolving bottlenecks. Swap exchanges orders between routes, improving load balance and efficiency.

VNS iteratively explores these neighborhoods. Upon improvement, the search restarts from 2-opt; otherwise, it proceeds to the next global step. This hierarchical strategy avoids local minima and enables thorough solution refinement. If progress stagnates (improvement < 0.1% over 30 iterations), the algorithm transitions from VNS to a more intensive local search strategy:

– Gradient-based Descent (GD) in a continuously relaxed solution space, with subsequent projection onto valid permutations;

– Fast 2-opt heuristic using incremental route-length updates.

*Step 7: Elitist Selection.* Retain 𝑁 best individuals. . For the next generation, the top N=80 individuals are selected.

To improve performance, we incorporate self-adaptation strategies. We apply a Self-adaptive Differential Evolution (SaDE) mechanism where the DE mutation factor *μ* and crossover rate (*CR)* are adapted every 50 generations based on their recent success rates. The search stops when relative improvement in the best solution’s fitness is below 0.1% over the last 30 iterations or upon reaching a predefined maximum iteration or runtime limit.

This hybrid approach combines global search (DE and GA operators) and intensive local improvement (LS), efficiently handling the multiple constraints characteristic of agricultural transport logistics while achieving cost-effective and environmentally sustainable route solutions.

Overall, the hybrid DE–GA+VNS algorithm successfully integrates multiple evolutionary operators and local search to handle the complex multi-constraint nature of the agricultural vehicle routing problem. Next, we describe computational experiments that evaluate the performance of this approach.

# **RESULTS AND DISCUSSION**

To evaluate the effectiveness of the proposed hybrid evolutionary algorithm (DE–GA+LS) and its variants, computational experiments were carried out using synthesized benchmark instances adapted specifically for agricultural logistics, incorporating key constraints: capacity, time windows, and LIFO loading requirements. The test instances were generated by modifying classic Solomon benchmarks (C101/RC206) [10], resulting in three distinct problem scales: a small instance (S-50, 50 orders and 10 vehicles), a medium instance (M-100, 100 orders and 15 vehicles), and a large instance (L-200, 200 orders and 25 vehicles). Each vehicle had a fixed cost of 120 u.c. (unit cost) per route, an 8t capacity, and an operational cost of 1.45 u.c./km.

All algorithms were executed for 10 independent runs per instance, with a computational budget limited to 500 generations. The CPU runtime was capped at 4 minutes for S-50, 8 minutes for M-100, and 18 minutes for L-200 instances. The performance of algorithms was evaluated based on several metrics: total logistics cost (Cost), deviation from the best-known solution (Gap %), computational runtime (Time), and solution stability (standard deviation, Stdev).

Table 1 summarizes the averaged results across all runs and instances. The baseline Genetic Algorithm (GA) achieved significantly higher costs across all instances, showing gaps of 14.6% (S-50), 17.4% (M-100), and 18.5% (L-200) relative to the best-known solutions. The baseline Differential Evolution (DE) improved upon GA performance, with cost gaps reduced to 7.3%, 9.4%, and 10.4%, respectively. Nevertheless, both baseline algorithms showed notably higher variability (Stdev values exceeding 500 for DE and 800 for GA), indicating less consistent solution quality.

**TABLE 1:** Cost, time, and stability metrics for GA, DE, and hybrid DE–GA algorithms with different local search strategies

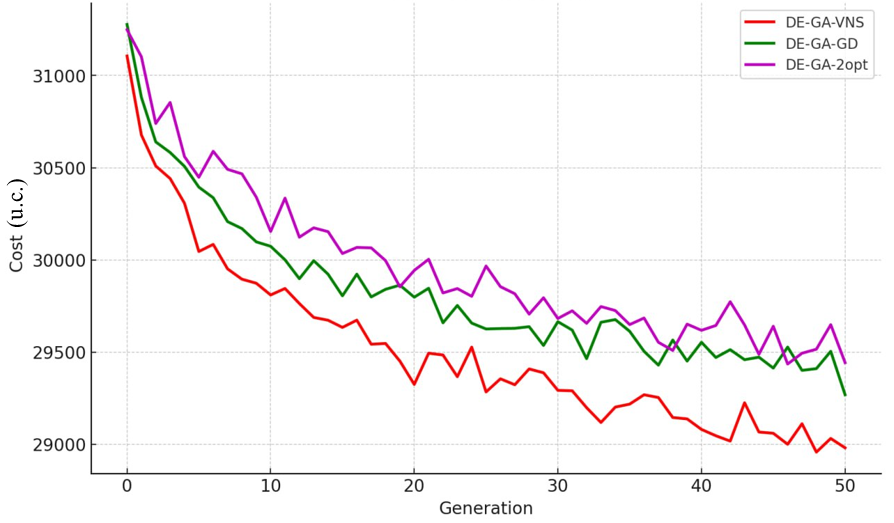
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **S-50 Cost** | **Gap (%)** | **Time (s)** | **Stdev** | **M-100 Cost** | **Gap (%)** | **Time (s)** | **Stdev** | **L-200 Cost** | **Gap %** | **Time, (s)** | **Stdev** |
| GA (baseline) | 32,850 | + 14.6 | 92 | 830 | 66,210 | + 17.4 | 196 | 1,490 | 134,700 | + 18.5 | 438 | 2,370 |
| DE (baseline) | 30,740 | +7.3 | 104 | 540 | 61,890 | +9.4 | 210 | 1,120 | 123,600 | + 10.4 | 462 | 1,750 |
| **DE-GA-VNS** | **27,930** | 0.0 | 118 | **380** | **56,780** | 0.0 | 244 | 690 | 110,820 | **0.0** | 529 | **1,090** |
| DE-GA-GD | 28,160 | +0.8 | 106 | 480 | 57,210 | +0.8 | 228 | 780 | 111,750 | +0.8 | 505 | 228 |
| DE-GA-2opt | 28,330 | + 1.4 | **88** | 450 | 57,460 | + 1.2 | 198 | 915 | 112,380 | + 1.4 | **467** | 1,470 |

In contrast, the primary proposed hybrid approach, DE-GA-VNS, demonstrated consistently superior performance. It produced the lowest total costs: 27,930 (S-50), 56,780 (M-100), and 110,820 (L-200), establishing it as the reference best solution (Gap % = 0). Additionally, DE-GA-VNS exhibited the lowest solution variability (standard deviation ranging from u.c.380 to u.c.1,090), highlighting its robustness and reliability across multiple runs.

Two simplified hybrid variants were tested alongside DE-GA-VNS. The first alternative, DE-GA-GD, replaced the VNS-based local search with a rank-based gradient descent, resulting in solutions only marginally inferior (≈0.8% higher costs on average). It provided a moderate gain in computational speed (≈10% reduction in runtime) compared to VNS, though with slightly increased variability (standard deviation about 25% higher).

The second variant, DE-GA-2opt, utilized a simplified and computationally efficient 2-opt heuristic for local search. This approach achieved the fastest average runtimes (approximately 25% faster than DE-GA-VNS) at the expense of a moderate deterioration in solution quality (≈ 1.4% higher cost) and increased variability (standard deviation around 40% higher).

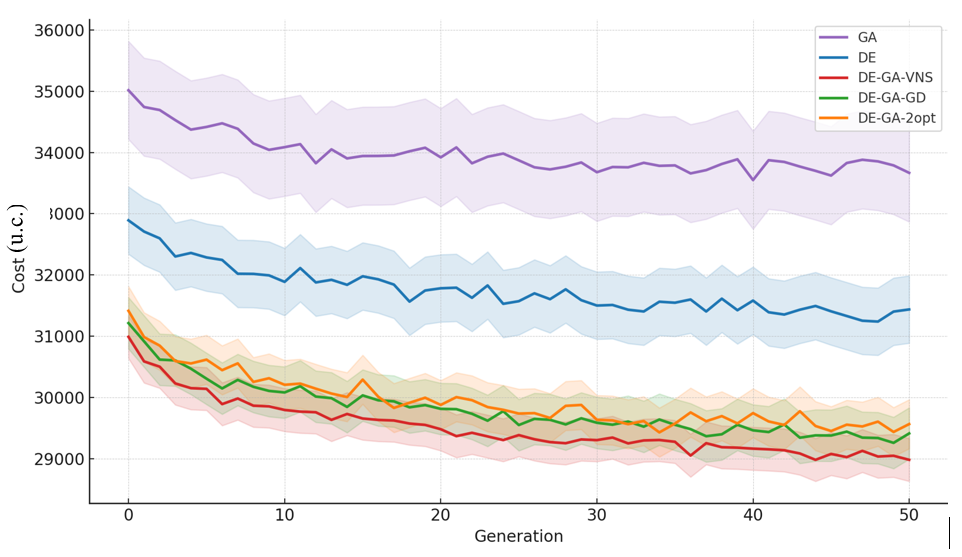
The convergence patterns of these algorithms, illustrated in Figures 1 and 2, clearly highlight the advantages of DE-GA-VNS. Figure 1 depicts the evolution of the total cost over 50 generations, demonstrating DE-GA-VNS consistently converging to the lowest-cost solution earlier in the search process. Figure 2 provides additional insight by including confidence intervals (standard deviation bands), illustrating lower variability for DE-GA-VNS solutions relative to GD and 2-opt alternatives, and a significant improvement over baseline methods.



**FIGURE 1.**  Cost convergence for algorithms over 50 generations

Statistical validation through paired t-tests further confirms the observed differences. Comparisons of DE-GA-VNS with baseline methods (GA and DE) indicated highly significant differences (p < 0.01) across all instances. Differences between DE-GA-VNS and the simplified variants (GD and 2-opt) remained statistically significant (p ≈ 0.02 and p ≈ 0.04, respectively), albeit with smaller effect sizes.

From a practical perspective, algorithm selection is context-dependent and should be guided by system-specific priorities. DE-GA-VNS, characterized by its lowest operational cost and highest robustness, is recommended for offline planning scenarios where minimizing total logistics cost, fuel consumption, emissions, and number of trips is paramount. DE-GA-GD, due to its computational structure compatible with GPU acceleration [15] and slightly reduced runtime, can be effectively applied to routing systems involving continuously defined or easily differentiable ranking variables. Lastly, DE-GA-2opt, featuring the simplest implementation and shortest computation times, proves particularly suitable for real-time dynamic route recalculations on mobile or edge-computing platforms, where computational speed is critical, and slight compromises in cost are acceptable.



**FIGURE 2.**  Cost convergence with variability bands (± standard deviation)

Overall, the experimental analysis confirms the capability of the proposed DE-GA-VNS hybrid evolutionary algorithm to deliver substantial cost savings (8–12% compared to baseline GA and 5–9% compared to baseline DE) while ensuring high stability and environmental sustainability. The reduction in total travel distances and the number of vehicle trips directly contributes to lower fuel consumption and emissions, aligning with modern sustainable transportation planning goals (as defined by cost equations (5)–(7) and observed convergence trends in Figures 1–2).

This work introduces a hybrid evolutionary algorithm for agricultural vehicle routing with capacity, time‑window, and LIFO loading constraints, minimizing the comprehensive cost F(π) (Equations (5)–(7)). Section 4 (Figures 1–2) shows that DE–GA–VNS attains the lowest costs and the highest stability across S‑50, M‑100, and L‑200. Cost reductions reach 8–12% vs. GA and 5–9% vs. DE, with no deadline violations. The gains follow from shorter total distance and improved load consolidation, which reduce trips, fuel use, and emissions.

Two streamlined local‑search hybrids quantify trade‑offs. DE–GA–GD (rank‑based descent with projection) is about 0.8% worse than VNS on average and ~10% faster, but more variable. DE–GA–2opt delivers the fastest convergence (~25% faster than VNS) at ≈1.3–1.4% higher costs and the largest variance. Paired t‑tests indicate p<0.01 versus GA/DE and p≈0.02/0.04 versus GD/2opt (Section 4). Relative to common two‑component hybrids, this scheme adds switchable LS variants triggered by stagnation and SaDE‑driven updates of *μ* and *CR*, which together sustain progress without sacrificing feasibility.

We apply DE operators in a real‑coded random‑key representation and decode (by sorting) to valid permutations with a feasibility repair step. GA retains effective subroutes; and VNS (2‑opt → relocate → swap) performs intensive local refinement. Although local search increases per‑generation work, fewer generations are typically needed, so wall‑clock times remain competitive or better. In practice: choose DE–GA–VNS for offline planning with maximum cost and sustainability gains; DE–GA–2opt for real‑time/edge re‑optimization with minimal latency; DE–GA–GD when rank‑based encodings or GPU acceleration are advantageous.

The experiments use static instances; the same code extends to rolling‑horizon or event‑driven updates, and to multi‑objective formulations that weight emissions alongside cost. Minimizing *F*(*π*) already reduces distance, trips, and fuel, reinforcing the environmental benefits visible in Figures 1–2.

# **CONCLUSION**

We developed and evaluated a hybrid evolutionary algorithm for agricultural vehicle routing with capacity, time‑window, and LIFO loading constraints (CVRPTW‑LIFO), minimizing the comprehensive cost 𝐹(𝜋). The method differs from common two‑component hybrids (DE+GA or GA+VNS) in three ways. Switchable local‑search variants—VNS (2‑opt → relocate → swap), a rank‑based (random‑key) descent with projection to permutations, and a fast 2‑opt—activated by a stagnation trigger. Self‑adaptive DE parameters (*μ, CR*) via a SaDE scheme. A real‑coded random‑key representation for applying DE in a continuous space with decoding (sorting) and feasibility repair back to valid routes.

This design sustains global exploration and intensive local refinement while respecting CVRPTW‑LIFO feasibility. Computational experiments (Section 4) show consistent cost improvements: 8–12% vs. GA and 5–9% vs. DE on instances up to 200 orders, with no deadline violations and low run‑to‑run variance. The full DE–GA–VNS variant yields the best costs and stability; DE–GA–2opt is ≈25% faster with ≤1.5% loss in cost; DE–GA–GD is ≈10% faster with ≈0.8% loss. The cost reductions arise from shorter total distance and better load consolidation, which also reduce the number of trips, fuel consumption, and emissions ‒supporting environmentally sustainable transport planning.

The algorithm scaled to larger instances without excessive runtime, indicating practicality for integration into transport planning systems. The same architecture extends naturally to rolling‑horizon/event‑driven re‑optimization and to multi‑objective formulations that explicitly weight ecological indicators (e.g., emissions) alongside cost. In sum, the proposed hybrid approach provides a robust, implementable pathway to cost‑efficient and sustainability‑oriented transport design in agricultural logistics.

# **REFERENCES**

1. K. Sethanan and T. Jamrus, *Comput. Ind. Eng.* **147**, 106640 (2020). doi:10.1016/j.cie.2020.106571.
2. B. Alhijawi and A. Awajan, *Evol. Intell.* **2024**(3) (2024).
3. E. Benavent, M. Landete, E. Mota, and G. Tirado, *Eur. J. Oper. Res.* **243**(3), 752–762 (2015). doi:10.1016/j.ejor.2014.12.029.
4. M. E. H. Sadati and B. Çatay, *Transp. Res. E Logist. Transp. Rev.* **149**, 102293 (2021).
5. O. Dib, L. Moalic, M. A. Manier, and A. Caminada, *Expert Syst. Appl.* **72**, 67–82 (2017). doi:10.1016/j.eswa.2016.12.009.
6. E. Cao, M. Lai, and K. Nie, in *Proc. 17th IFAC World Congress*, Seoul, Korea, 2008, p. 10576–10581. doi:10.3182/20080706-5-KR-1001.01791.
7. L. F. Sulyukova and Z. I. Akhmedjanova, *E3S Web Conf.* **401**, 05011 (2023). doi:10.1051/e3sconf/202340105011.
8. H. Pollaris, K. Braekers, A. Caris, G. K. Janssens, and S. Limbourg, *OR Spectr.* **37**(2), 297–330 (2015).
9. J. F. Cordeau, G. Laporte, M. W. Savelsbergh, and D. Vigo, in *Handbooks in Operations Research and Management Science*, Vol. 14, edited by C. Barnhart and G. Laporte (Elsevier, Amsterdam, 2007), p. 367–428.
10. H. Gehring and J. Homberger, *Asia Pac. J. Oper. Res.* **18**(1), 35–47 (2001).
11. E. G. Talbi, *Metaheuristics: From Design to Implementation* (Wiley, Hoboken, NJ, 2009), 624 p.
12. A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing* (Springer, Berlin, 2003), 300 p.
13. D. J. Bowersox, D. J. Closs, and M. B. Cooper, *Supply Chain Logistics Management* (McGraw-Hill, New York, 2013), 484 p.
14. J. Sripriya, A. Ramalingam, and K. Rajeswari, in *Proc. 2015 Int. Conf. Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 2015, p. 1–4. doi:10.1109/ICIIECS.2015.7193072.
15. Q. M. Alzubi, M. Anbar, Y. Sanjalawe, M. A. Al-Betar, and R. Abdullah, *Expert Syst. Appl.* **204**, 117597 (2022). doi:10.1016/j.eswa.2022.117597.