Solving Electric Vehicle Routing by Iterative Instance Refinement Framework

Tzu-Hao Lin
Department of Computer Science
National Yang Ming Chiao Tung University
HsinChu City, TAIWAN
kartd0820.cs13@nycu.edu.tw

Ying-ping Chen*
Department of Computer Science
National Yang Ming Chiao Tung University
HsinChu City, TAIWAN
ypchen@cs.nycu.edu.tw

***Abstract*—The Electric Vehicle Routing Problem (EVRP) requires the joint optimization of customer routing and energy feasibility under battery capacity constraints. Although many existing EVRP solution strategies implicitly follow a bilevel structure, routing and feasibility handling are commonly implemented within monolithic solvers, making algorithmic components tightly coupled and difficult to substitute or reuse.**

This paper introduces an Iterative Instance Refinement Framework (IIRF) that explicitly decouples routing optimization from feasibility enforcement through a problem-instance interface. The framework links an upper-level VRP solver with a lower-level EV feasibility evaluator via an EV-aware cost matrix to iteratively refine the problem instance for steering the routing solver toward energy-feasible solutions. By construction, the framework is both solver-agnostic and evaluator-agnostic, enabling different routing algorithms, feasibility evaluators, and refinement strategies to be integrated without altering the algorithmic core.

Empirical results show that refining problem instances enables a classical VRP solver to produce high-quality and energy-feasible solutions without explicitly handling charging decisions during its search. On two EVRP benchmark suites, E-X and E-F-M-X, IIRF not only delivers the state-of-the-art performance but also advances the best-known solutions to date for multiple problems. These findings validate the feasibility of the modular design and indicate a pathway toward an evolutionary optimization framework that operates at the level of problem instances, rather than exclusively at the level of explicit solution representations.

***Index Terms*—Evolutionary computation, Combinatorial optimization, Electric vehicle routing problem, Instance space evolution, Iterative instance refinement framework.**

I. Introduction

The Electric Vehicle Routing Problem (EVRP) has gained increasing importance due to the growing adoption of electric vehicles in logistics and transportation [1]. Beyond classical routing and capacity constraints, the EVRP explicitly incorporates battery limitations and charging operations, essential for the practical deployment of electric fleets. From an optimization standpoint, these extra constraints introduce strong dependencies between routing decisions and energy feasibility, thereby substantially increasing problem complexity [2].

Approaches for the EVRP generally fall into three categories: exact, heuristic, and hyperheuristic methods. Exact methods typically formulate the EVRP as a mixed-integer linear programming model and provide optimality guarantees for small-sized instances, but suffer from exponential runtime growth as problem size increases, limiting their practicality. Heuristic methods represent the dominant strategy for practical EVRPs, trading optimality for efficiency. Representatives include variable neighborhood search (VNS) [3], ant colony optimization (ACO) [4], genetic algorithms (GA) [5], and iterative local search (ILS) [6]. Hyperheuristic methods operate by selecting, combining, or adapting multiple low-level heuristics and demonstrate strong performance on large EVRP instances [7], [8]. While these methods achieve good empirical performance, most EVRP heuristics are designed for a specific variant and thus lack the flexibility to support multiple EVRP formulations within a unified framework. Recent studies incorporate learning mechanisms [9] into evolutionary algorithms to enhance generality across EVRP variants. Our work shares a similar motivation but approaches the problem from a different perspective: instead of improving the adaptability of the search process as in learning-based evolutionary methods, we focus on the algorithmic architecture and observe that most EVRP heuristics remain monolithic, tightly coupling routing, feasibility handling, and charging decisions.

Motivated by these considerations, this study adopts a modular perspective and investigates how decoupling routing optimization, feasibility evaluation, and cost representation can enhance flexibility and analytical clarity in EVRP solver design. Beyond the EVRP domain, this perspective suggests a more general evolutionary optimization paradigm in which the evolutionary search operates through problem-instance refinement rather than exclusively through explicit solution manipulation. From a broader perspective, this design introduces an instance-space search axis that complements classical solution-space evolution, wherein feasibility information refines the problem instance rather than constraining variation operators. This aligns with a more general framework design philosophy favored in evolutionary computation, where problem components can be modularly replaced, combined, and analyzed. The main contributions of this work are summarized as follows:

- We introduce the **Iterative Instance Refinement Framework (IIRF)**, a modular framework for constrained optimization in which feasibility iteratively refines the problem instance rather than directly modifying solutions.
- We instantiate the IIRF in the EVRP domain, forming IIRF-EVRP, in which routing optimization and energy

feasibility are handled by the upper and lower levels, respectively. The instantiation is both solver-agnostic and evaluator-agnostic, making the routing solver and feasibility evaluator exchangeable components. Since the two modules interact through a problem-instance interface, additional EVRP constraints, e.g., time windows, heterogeneous fleets, or battery models, can be handled by either the solver or the evaluator without modifying the overall algorithmic architecture.
- We develop an **EV-Aware Instance Refinement** mechanism in which the cost representation serves as the interface for feasibility feedback, enabling iterative modification of the problem instance without altering either the routing solver or the feasibility evaluator.

The remainder of this paper is organized as follows. Section II reviews related work and motivates the modular perspective adopted in this study. Section III formalizes the EVRP. Section IV introduces the proposed framework. Section V presents its realization for EVRP. Section VI describes the experimental setup, and Section VII reports the results, followed by the conclusions drawn in Section VIII.

The solutions obtained in this study are publicly available at `https://github.com/nclab/iirf-evrp` .

II. Related works

Existing approaches for the EVRP primarily operate at the level of solution representations, where routing plans are iteratively manipulated, and feasibility is restored through repair. Heuristic methods such as VNS, ACO, ILS, and GA remain widely used in practical EVRPs, providing effective exploration of routing spaces under capacity and charging constraints. However, these approaches are generally monolithic and variant-specific, tightly intertwining routing decisions with EV-related feasibility/constraint handling.

To improve the robustness of metaheuristic search and reduce manual parameter tuning, hyperheuristic and learning-assisted strategies have recently been introduced. Hyper-Heuristic Simulated Annealing (HHASA) [7] dynamically selects low-level search operators during optimization, reducing the burden of operator configuration. Q-learning-based hyper-heuristics [8] similarly employ reinforcement learning to adapt operator selection online. These approaches remain solution-driven, operating through direct solution manipulation, and are complementary to modular decomposition strategies.

A parallel line of research explicitly separates routing from energy-feasibility handling through bilevel designs. The Bilevel ACO (BACO) method [10] was the first to formulate the EVRP as a bilevel optimization problem, decomposing routing decisions at the upper level and feasibility restoration through the Fixed-Route Vehicle Charging Problem (FRVCP) at the lower level. Confidence-Based ACO (CBACO) [11] further introduced confidence-based filtering to reduce the number of candidate solutions forwarded to the FRVCP evaluator. More recently, the Bilevel Hybrid Genetic Algorithm (BHGA) [12] combined a GA-based upper-level solver with a feasibility restoration procedure and achieved state-of-the-art performance on the CEVRP benchmark [13], updating multiple best-known solutions.

Surrogate-assisted evolutionary approaches form a closely related direction. The Surrogate-Accelerated Evolutionary Algorithm (SAEA) [9] incorporates learned surrogates to approximate EV-feasible route cost, reducing reliance on lower-level FRVCP evaluation and enabling larger-scale evolutionary search than bilevel methods that evaluate feasibility exactly.

Across these methods, feasibility handling is fundamentally implemented as a solution-driven repair mechanism applied to candidate routing plans. Consequently, routing optimization and energy feasibility are executed within the same evolutionary loop, preserving a tight computational coupling between the two components. This coupling limits solver modularity and reduces portability across EVRP variants, as routing, feasibility enforcement, and cost modeling remain fused within a single algorithmic design.

Motivated by these observations, the present work explores an alternative design paradigm in which feasibility feedback is repurposed as information for refining the problem instance. Rather than directly modifying solutions as routing representations, feasibility acts as a driver for instance-level refinement, enabling routing optimization and EV-specific feasibility handling to be modularly decoupled.

III. The Electric Vehicle Routing Problem

We consider the *Electric Vehicle Routing Problem* (EVRP), a variant of the Vehicle Routing Problem (VRP) that explicitly accounts for the limited battery capacity of electric vehicles (EVs) and the availability of charging stations.

The problem is defined on a complete weighted undirected graph $G = (V, E)$, where the node set $V = \{0\} \cup I \cup F$ consists of a single central depot $\{0\}$, a set of customers I, and a set of charging stations F. Each edge $(i, j) \in E$ is associated with a non-negative distance d_{ij}. For formulation, F' is defined as an expansion of F, which contains $2|I|$ copies of each charging station. Each customer $i \in I$ has a positive demand q_i that must be satisfied exactly once. All electric vehicles (EVs) are homogeneous, with a maximum cargo capacity C and a maximum battery capacity Q. Vehicles depart from the depot fully loaded and fully charged, and must return to the depot at the end of their routes. Energy consumption is assumed to be proportional to the traveled distance. Specifically, traversing edge (i, j) consumes $e_{ij} = h \cdot d_{ij}$ units of energy, where h is a constant energy consumption rate.

The EVRP is subject to the following constraints:

- **Depot constraint:** All vehicles start and end their routes at the depot, departing fully loaded and fully charged.
- **Customer service constraint:** Each customer must be visited exactly once by a vehicle, and its demand must be fully satisfied.
- **Vehicle capacity constraint:** The total demand served along any route must not exceed the vehicle cargo capacity C.
- **Energy feasibility constraint:** The battery level of a vehicle must never drop below zero along its route.

- **Recharging constraint:** Vehicles may visit charging stations multiple times, where the battery is fully recharged up to the maximum battery capacity Q.

The objective of the EVRP is to minimize the total distance traveled by all vehicles while satisfying customer demands, vehicle capacity constraints, and energy feasibility constraints through optional visits to charging stations.

The EVRP math model is formulated as follows by [13]:

$$\min \sum_{i\in V,\, j\in V,\, j\neq i} d_{ij}x_{ij} \tag{1}$$

s.t.

$$\sum_{j\in V,\, j\neq i} x_{ij} = 1,\ \forall i \in I, \tag{2}$$

$$\sum_{j\in V,\, j\neq i} x_{ij} \leq 1,\ \forall i \in F', \tag{3}$$

$$\sum_{j\in V,\, j\neq i} x_{ij} - \sum_{j\in V,\, j\neq i} x_{ji} = 0,\ \forall i \in V, \tag{4}$$

$$u_j \leq u_i - q_i x_{ij} + C(1 - x_{ij}),\ \forall i, j \in V,\ i \neq j, \tag{5}$$

$$0 \leq u_i \leq C,\ \forall i \in V, \tag{6}$$

$$y_j \leq y_i - hd_{ij}x_{ij} + Q(1 - x_{ij}), \forall i \in I, \forall j \in V,\ i \neq j, \tag{7}$$

$$y_j \leq Q - hd_{ij}x_{ij},\ \forall i \in F' \cup \{0\},\ \forall j \in V,\ i \neq j, \tag{8}$$

$$0 \leq y_i \leq Q,\ \forall i \in V, \tag{9}$$

$$x_{ij} \in \{0, 1\},\ \forall i, j \in V,\ i \neq j, \tag{10}$$

where Eq. (1) defines the objective function of the EVRP, which aims to minimize the total travel distance of all EVs. Constraint (2) enforces that each customer is served exactly once. Constraint (3) allows charging stations to be visited for recharging. Constraint (4) enforces flow conservation at each node. Constraints (5) and (6) impose the vehicle capacity limits. Constraints (7)–(9) ensure energy feasibility by guaranteeing that the battery level never drops below zero. Finally, Constraint (10) defines the binary decision variable x_{ij}, which takes the value 1 if arc (i, j) is traveled by an EV, and 0 otherwise. Variables u_i and y_i denote, respectively, the remaining cargo capacity and remaining battery charge of an EV upon reaching node $i \in V$.

IV. ITERATIVE INSTANCE REFINEMENT FRAMEWORK

Common evolutionary optimization operates over a solution space. Let $\mathcal{X}$ denote the search space, i.e., the set of all possible solutions, associated with a given optimization problem. Broadly, evolutionary optimization encompasses population-based evolutionary algorithms (e.g., GA, ES), swarm and constructive metaheuristics (e.g., ACO, PSO), and trajectory-based local improvement methods (e.g., VNS, ILS). Despite algorithmic differences, these methods share a common feature: they evolve candidate solutions for a fixed underlying.

More formally, classical evolutionary algorithms maintain a population $P_k \subseteq \mathcal{X}$ at iteration/generation k and update it through selection and variation operators:

$$P_{k+1} = \mathcal{A}(P_k), \qquad P_k \subseteq \mathcal{X}, \tag{11}$$

where $\mathcal{A}(\cdot)$ denotes the transformation composed by the algorithmic operators, including selection, crossover, mutation, and others. A representative solution $x_k \in P_k$ (e.g., the current best or an elite individual) is typically used to characterize the progress of the search.

For problems with simple feasibility structures, such solution-centric evolution is effective. However, in constrained combinatorial domains with intertwined feasibility requirements—such as the EVRP, where routing, energy, charging, and capacity constraints interact—feasibility is not a local property of the solution representation. Designing feasibility-preserving variation operators becomes costly, and repair may require sophisticated variant-specific procedures, limiting scalability and portability across problem variants.

These observations motivate shifting part of the optimization burden from the solution space to the instance space. Instead of enforcing feasibility solely through solution-space operators, feasibility information can be externalized and incorporated to navigate in the space of problem instances.

To formalize this idea, let $\mathcal{I}_0$ denote the original problem instance. Since direct optimization over $\mathcal{I}_0$ may require specialized feasibility-handling mechanisms, we instead project $\mathcal{I}_0$ into an auxiliary instance space $\mathcal{P}$ that admits efficient optimization, yielding an initial projected instance $\mathcal{P}_0 \in \mathcal{P}$.

Given a projected instance $\mathcal{P}_k$, an optimizer computes

$$x_k = \arg\min_{x\in\mathcal{X}} f(x; \mathcal{P}_k), \tag{12}$$

where $f(\cdot; \mathcal{P}_k)$ denotes the objective induced by the current projected instance. The solution x_k is then evaluated against the original problem instance $\mathcal{I}_0$ through an external evaluator

$$s_k = \mathcal{E}(x_k; \mathcal{I}_0), \tag{13}$$

which returns a feasibility signal s_k describing constraint satisfaction or violation. Based on this signal, a refinement operator updates the projected instance:

$$\mathcal{P}_{k+1} = \Phi(\mathcal{P}_k, s_k), \tag{14}$$

inducing a trajectory

$$\mathcal{P}_0 \to \mathcal{P}_1 \to \cdots \to \mathcal{P}_K. \tag{15}$$

Instance-space evolutionary perspective. This formulation yields an instance-space evolutionary viewpoint: feasibility guidance is injected into the search not by repairing or constraining solutions in $\mathcal{X}$, but by refining the projected instance in $\mathcal{P}$. This instance-space interpretation decouples feasibility handling from solution variation, enabling evolutionary search to operate without encoding constraint-handling or feasibility-restoration procedures into solution operators. In contrast, bilevel approaches retain feasibility as a solution-coupled procedure that is invoked on solutions rather than absorbed into the problem instance.

V. FRAMEWORK REALIZATION FOR EVRP

To instantiate IIRF for the EVRP, we construct IIRF-EVRP comprising three functional components: a routing optimizer, a feasibility evaluator, and an instance refinement mechanism.

These components correspond respectively to the optimizer in Eq. (12), the evaluator $\mathcal{E}(\cdot)$ in Eq. (13), and the refinement operator $\Phi(\cdot)$ in Eq. (14). In what follows, we describe how these components are concretely realized for the EVRP.

A. Routing Optimizer

The routing optimizer is instantiated using PyVRP, an open-source Python solver for vehicle routing problems. PyVRP adopts Hybrid Genetic Search (HGS), originally proposed in [14] and later realized as an efficient Python-based solver in [15]. This choice is consistent with the design of IIRF, as the routing optimizer operates without EV-specific constraints.

B. Feasibility Evaluator

The feasibility evaluator is instantiated using an exact approach to the Fixed Route Vehicle Charging Problem (FRVCP). The underlying labeling-based algorithm was proposed in [16] and later implemented as an open-source Python solver, frvcpy [17], permitting direct integration into IIRF. This feasibility evaluator enforces battery and charging constraints with respect to the original EVRP instance, while preserving the customer visit sequence produced by the routing optimizer.

C. Instance Refinement

To glue the routing optimizer and the feasibility evaluator, we introduce an EV-aware instance refinement mechanism in which the cost matrix serves as the sole interface between them. The routing optimizer queries the matrix to generate candidate routes, while the feasibility evaluator processes such routes and returns feasibility feedback extracted from EV-feasible solutions. Instead of modifying the routing decisions directly, feasibility information is incorporated into the instance representation via updating the cost matrix, allowing feasibility as guidance to influence the routing search.

1) Cost Matrix Decomposition: To support instance refinement, for the routing optimizer, the cost matrix is decomposed:

$$D = D_{\text{base}} + \Delta, \tag{16}$$

where D_{base} provides a static EV-aware approximation of energy feasibility, and Δ denotes a learned residual matrix incorporating feasibility corrections. Initially, $\Delta = 0$ and is updated during refinement iterations.

2) Base EV-Aware Cost Matrix: The base matrix D_{base} approximates energy-feasibility constraints without embedding battery dynamics within the routing optimizer. For each location pair i, j, the updated cost is defined as

$$\begin{aligned} D_{\text{base}}(i,j) = (1-\alpha)\, d(i,j) + \alpha \min_{s \in F}\{d(i,s) + d(s,j)\} \\ + \lambda \max\{0,\, d(i,j) - R\}, \end{aligned} \tag{17}$$

where $i, j \in \{0\} \cup I$ denote the depot and customer nodes, $d(\cdot,\cdot)$ denotes the distance between two locations, F is the set of charging stations, and $R = Q/h$ represents an estimate range threshold derived from battery capacity Q and energy consumption rate h. The first term preserves geometric VRP structure, the second reflects the shortest charging detour, and the third penalizes arcs whose direct length exceeds the threshold. The parameters α and λ are tunable parameters. R, Q, h, and F are obtained directly from the EVRP instance.

3) Learned Residual Matrix: The residual matrix Δ captures feasibility corrections obtained from the feasibility evaluator and modifies the cost matrix by adjusting arc costs to account for energy-related violations in EV-feasible routes.

4) Feasibility Residual Extraction: After feasibility evaluation, charging stations may be inserted along the route to satisfy battery constraints. These insertions induce additional travel cost relative to the original customer sequence. For two consecutive customers u and v in the original route, let $\tilde{x}_{u \to v}$ denote the charging-augmented subpath returned by the feasibility evaluator. The positive residual is defined as

$$r(u,v) = \max\big(0,\, C_{\text{true}}(\tilde{x}_{u \to v}) - C_{\text{direct}}(u,v)\big), \tag{18}$$

where $C_{\text{direct}}(u,v)$ denotes the cost under the current effective cost matrix D, and $C_{\text{true}}(\cdot)$ denotes the executable EV-feasible cost under the original EVRP instance. Positive residuals indicate customer transitions that systematically require charging detours or incur additional overhead to remain feasible.

5) Residual-Based Instance Refinement: Residual feedback is then incorporated by updating the residual matrix Δ, and the routing optimizer evaluates the effective cost matrix D in the next iteration. For a set of observed residuals $\{r(u,v)\}$, the refinement mechanism applies exponential smoothing as

$$\Delta(u,v) \leftarrow (1-\eta)\, \Delta(u,v) + \eta\, r(u,v), \tag{19}$$

where $\eta \in (0,1)$ is a learning rate controlling the influence of newly observed residuals. To discount outdated information, the residual matrix further undergoes decay as

$$\Delta \leftarrow \gamma\, \Delta, \tag{20}$$

where $\gamma \in (0,1)$ controls temporal forgetting. The refinement operator penalizes customer transitions that repeatedly induce charging detours or energy overhead, progressively steering the routing optimizer toward energy-feasible solution structures.

6) Stabilization Mechanisms: Two stabilizing mechanisms are incorporated into the refinement process to enhance numerical stability and prevent over-penalization. First, residual values are clipped at a bound $r_{\max}$ to prevent extreme charging detours from producing unbounded updates. Second, if the EV-feasible cost deteriorates relative to the previous iteration, the residual matrix is shrunk by a factor $\beta \in (0,1)$, preventing the refinement from locking onto misleading penalty structures. Algorithm 1 presents the full EV-aware instance refinement, including residual extraction, matrix updates, and stabilization.

Support for EVRP Variants EV-related variants introduce additional constraints that can be localized to either the feasibility evaluator or the routing optimizer. Variants involving charging behavior, e.g., partial charging, heterogeneous charging stations, can be supported by using suitable feasibility evaluators. Variants related to route construction, e.g., multi-depot or heterogeneous fleets, require routing solvers with corresponding capabilities. Time-window variants need both. This modular design leverages existing advances from VRP and FRVCP research; for instance, PyVRP supports multi-depot, time windows, and heterogeneous fleets, whereas frvcpy supports partial charging and heterogeneous charging stations.

Algorithm 1 EV-Aware Instance Refinement

```
Input: EVRP instance; routing solver; feasibility evaluator
Parameters: Θ = {α, λ, η, γ, r_max, β, T}
Compute static EV-aware cost matrix D_base using Eq. (17)
Initialize Δ ← 0
for t = 1 to T do
    D ← D_base + Δ
    R ← SolveRouting(D)
    r(u,v) ← EvaluateFeasibility(R)
    r(u,v) ← min(r(u,v), r_max)                    ▷ clip
    for each (u,v) with r(u,v) > 0 do
        Δ(u,v) ← (1 − η)Δ(u,v) + η r(u,v)
    end for
    Δ ← γ Δ                                        ▷ decay
    if feasible cost worsens then
        Δ ← β Δ                                    ▷ shrink
    end if
end for
```

VI. Experimental Setup

A. Benchmark Description

In this study, test cases are categorized into two scales: *small-scale* instances with up to 100 customers, and *large-scale* instances with more than 100 and up to 1000 customers.

The first benchmark is the *E-X* suite [13], developed for the IEEE WCCI 2020 competition. It contains 17 instances, 7 small-scale and 10 large-scale cases. These instances extend classical CVRP datasets to the electric setting by adding charging requirements, where charging stations are arranged over a 9-grid or 16-grid spatial layout to enforce intermediate recharging and expose solvers to battery constraints.

The second benchmark is the *E-F-M-X* suite [18], introduced as an enhanced infrastructure model. Instead of grid layouts, charging stations are placed such that every customer lies within a specified coverage radius of at least one station, improving realism and aligning with urban deployment scenarios. The E-F-M-X suite comprises 24 instances, 10 small-scale and 14 large-scale cases.

B. Baseline Algorithms

To enable a meaningful comparison across the two benchmark suites, we adopt the baseline selection used by SAEA [9], one of the few studies reporting results on both E-X and E-F-M-X. Under this setting, ILS, TS, SIGALNS, SA, VNS, GA, BACO, and SAEA are adopted for the E-X suite. For the E-F-M-X suite, ILS, SIGALNS, Max–Min Ant System (MMAS), Memetic Algorithm (MA), BACO, Confidence-Based BACO (CBACO), and SAEA are adopted.

Since the majority of these algorithms do not provide open source or executable binaries, their performance results included in this study are directly from [9]. It is noted that several other EVRP methods available in the literature achieve competitive performance. Those methods are not included for comparison because they either adopt different evaluation protocol, such as fewer or more function evaluations [7], [12], [19], do not report results on the full benchmark suites [20], or use execution time as the termination criterion rather than evaluation budgets [8], [21], preventing consistent comparison.

C. Experimental Configuration

We follow the experimental configuration used in [9] for benchmarking. Each problem instance is run for 20 independent trials, and the maximum number of objective evaluations is set to 25,000 per trial.

For the proposed method, the number of IIRF iterations is set to $T = 50$. For each iteration, the routing optimizer performs 499 evaluations with the current cost matrix, and the feasibility evaluator executes one exact EV-feasibility evaluation. This yields a total of $500 \times 50 = 25{,}000$ evaluations per trial, matching the specified evaluation budget.

The EV-Aware Instance Refinement mechanism is instantiated with the following parameter settings. The base cost matrix parameters are set to $\alpha = 0.05$ and $\lambda = 4.0$. The learning parameters for residual update are set to $\eta = 0.08$, $r_{\max} = 500.0$, $\gamma = 0.95$, and $\beta = 0.90$.

VII. Results and Discussion

This section provides the experimental results from different perspectives. First, we report the numerical performance obtained of the proposed method and baseline algorithms. Then, we conduct a non-parametric statistical analysis to rank the algorithms as well as to assess the statistical significance.

A. Numerical Results

The numerical results are reported in Tables II and III. The best value among all algorithms in comparison is highlighted in bold. For the proposed IIRF, the minimum value is annotated as (1) an asterisk ($*$) indicates that the value matches the best result reported in the literature, while two asterisks ($**$) indicate that the value advances the best-known result to date. The best-known results are collected from the baseline algorithms and also from the empirical results reported in studies [7], [8], [12], [21], to the best of our knowledge.

Tables II and III report the numerical results on the E-X and E-F-M-X suites, respectively. For E-X, IIRF achieves the best mean performance on 10 out of the 17 instances among all baseline algorithms, 9 of which are large-scale cases, indicating that IIRF performs particularly well on larger instances. For E-F-M-X, the more challenging, realistic benchmark, IIRF achieves the best mean performance on 19 out of the 24 instances. In particular, it obtains the best mean on 13 out of the 14 large-scale instances. Regarding the best-known values, IIRF advances the state-of-the-art performance on four instances, namely M-n212-k16-s12, X-n147-k7-s4, X-n221-k11-s7, and X-n830-k171-s11. These results indicate that IIRF is especially competitive on the more difficult problems and performs on par with the leading algorithms in existence.

B. Non-parametric Analysis

To assess the statistical significance of the performance difference, we first apply the Friedman test to examine whether the null hypothesis of equal performance can be rejected. Upon rejection, a post-hoc pairwise hypothesis testing procedure

TABLE I: Average Friedman rankings and Holm p-values ($\alpha = 0.05$) for the E-X and E-F-M-X benchmark suites

Algorithm	Ranking	p_{Holm}
IIRF	2.5294	
BACO	2.6176	0.925162
VNS	3.5294	0.574131
SA	4.6471	0.072512
TS	5.1765	0.019329
SAEA	5.5588	0.006297
GA	5.7059	0.004328
SIGALNS	7.6176	$< 10^{-6}$
ILS	7.6176	$< 10^{-6}$

(a) E-X

Algorithm	Ranking	p_{Holm}
IIRF	1.4583	
SAEA	2.8125	0.055482
BACO	3.3542	0.014676
MA	3.7292	0.000396
CBACO	4.8125	0.000008
ILS	6.0417	$< 10^{-6}$
MMAS	6.4583	$< 10^{-6}$
SIGALNS	7.3333	$< 10^{-6}$

(b) E-F-M-X

is performed to determine whether IIRF significantly outperforms the remaining algorithms. In the post-hoc analysis, IIRF is used as the control algorithm. This procedure is separately applied to the E-X and E-F-M-X benchmark suites.

Table I reports the results for the E-X and the E-F-M-X benchmark suites. In Table I(a), the Friedman test indicates that the performance differences among the algorithms are statistically significant with $p < 10^{-10}$. Using IIRF as the control algorithm, the Holm post-hoc results show that IIRF significantly outperforms GA, ILS, TS, SIGALNS, and SAEA for $\alpha = 0.05$. On the other hand, the performance differences between IIRF and BACO, VNS, and SA are not statistically significant, indicating that these three methods are the closest competitors to IIRF on E-X. In Table I(b), the Friedman test indicates that the performance differences among the methods are statistically significant with $p < 10^{-20}$. Using IIRF as the control algorithm, the Holm post-hoc results show that IIRF significantly outperforms ILS, SIGALNS, MMAS, MA, BACO, and CBACO for $\alpha = 0.05$, while the difference between IIRF and SAEA is not statistically significant ($p = 0.055$), making SAEA the closest competitor on E-F-M-X.

VIII. Conclusions and Future Work

In this work, we introduced the iterative instance refinement framework, IIRF, and instantiated IIRF-EVRP for the EVRP. By decoupling routing optimization and feasibility enforcement, IIRF enables a VRP solver to work without EV-specific mechanisms. The empirical results show that IIRF delivers superior performance, particularly on the larger, more challenging cases. In addition, IIRF advances several the best-known results to date, further demonstrating its effectiveness. Future research includes exploring more sophisticated refinement mechanisms and extending IIRF to other combinatorial optimization problems. Finally, understanding how instance-space evolution interacts with classical evolutionary operators and search strategies remains an open and intriguing question.

Acknowledgments

The work was supported in part by the National Science and Technology Council of Taiwan under Grant NSTC 114-2221-E-A49-094.

References

[1] I. Kucukoglu, R. Dewil, and D. Cattrysse, "The electric vehicle routing problem and its variations: A literature review," *Computers & Industrial Engineering*, vol. 161, p. 107650, 2021.

[2] C. Lin, K. Choy, G. Ho, S. Chung, and H. Lam, "Survey of green vehicle routing problem: Past and future trends," *Expert Systems with Applications*, vol. 41, no. 4, Part 1, pp. 1118–1138, 2014.

[3] A. Verma, "Electric vehicle routing problem with time windows, recharging stations and battery swapping stations," *EURO Journal on Transportation and Logistics*, vol. 7, no. 4, pp. 415–451, 2018.

[4] M. Mavrovouniotis, G. Ellinas, and M. Polycarpou, "Ant colony optimization for the electric vehicle routing problem," in *2018 IEEE Symposium Series on Computational Intelligence*, 2018, pp. 1234–1241.

[5] Z. Guo, Y. Li, X. Jiang, and S. Gao, "The electric vehicle routing problem with time windows using genetic algorithm," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2017, pp. 635–639.

[6] O. Sassi, W. R. Cherif-Khettaf, and A. Oulamara, "Multi-start iterated local search for the mixed fleet vehicle routing problem with heterogenous electric vehicles," in *Evolutionary Computation in Combinatorial Optimization*. Springer International Publishing, 2015, pp. 138–149.

[7] E. Rodríguez-Esparza, A. D. Masegosa, D. Oliva, and E. Onieva, "A new hyper-heuristic based on adaptive simulated annealing and reinforcement learning for the capacitated electric vehicle routing problem," *Expert Systems with Applications*, vol. 252, p. 124197, 2024.

[8] J. Lin, X. Wang, R. Niu, and Y. He, "A Q-learning-based hyper-heuristic for capacitated electric vehicle routing problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 10, pp. 15 746–15 757, 2025.

[9] C. Wang, S. Ma, B. Zha, and B. Ma, "Surrogate-accelerated evolutionary algorithm for solving electric vehicle routing problems," *IEEE Trans. Transp. Electrific.*, vol. 11, no. 1, pp. 3524–3537, 2025.

[10] Y.-H. Jia, Y. Mei, and M. Zhang, "A bilevel ant colony optimization algorithm for capacitated electric vehicle routing problem," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10 855–10 868, 2022.

[11] ——, "Confidence-based ant colony optimization for capacitated electric vehicle routing problem with comparison of different encoding schemes," *IEEE Trans. Evol. Comput.*, vol. 26, no. 6, pp. 1394–1408, 2022.

[12] C.-T. Feng, Y.-H. Jia, Q. Yang, W.-N. Chen, and H. Jiang, "A bilevel hybrid genetic algorithm for capacitated electric vehicle routing problem," in *2024 IEEE Congress on Evolutionary Computation (CEC)*, 2024.

[13] M. Mavrovouniotis, C. Menelaou, S. Timotheou, G. Ellinas, C. Panayiotou, and M. Polycarpou, "A benchmark test suite for the electric capacitated vehicle routing problem," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.

[14] T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei, "A hybrid genetic algorithm for multidepot and periodic vehicle routing problems," *Operations Research*, vol. 60, no. 3, pp. 611–624, 2012.

[15] N. A. Wouda, L. Lan, and W. Kool, "PyVRP: A high-performance VRP solver package," *INFORMS Journal on Computing*, vol. 36, no. 4, pp. 943—-955, Jul. 2024.

[16] A. Froger, J. E. Mendoza, O. Jabali, and G. Laporte, "Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions," *Computers & Operations Research*, vol. 104, pp. 256–294, 2019.

[17] N. Kullman, A. Froger, J. Mendoza, and J. Goodson, "frvcpy version v2020.1035," *INFORMS Journal on Computing*, 2020. [Online]. Available: https://github.com/INFORMSJoC/2020.1035

[18] M. Mavrovouniotis, C. Menelaou, S. Timotheou, G. Ellinas, C. Panayiotou, and M. Polycarpou, "A benchmark test suite for the electric capacitated vehicle routing problem," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1—-8.

[19] C. Wang, M. Cao, H. Jiang, X. Xiang, and X. Zhang, "A deep reinforcement learning-based adaptive large neighborhood search for capacitated electric vehicle routing problems," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 9, no. 1, pp. 131–144, 2025.

[20] S. Zhu and C. Wang, "An interaction-enhanced co-evolutionary algorithm for electric vehicle routing optimization," *Applied Soft Computing*, vol. 165, p. 112113, 2024.

[21] Y. Chen, J. Xue, Y. Zhou, and Q. Wu, "An efficient threshold acceptance-based multi-layer search algorithm for capacitated electric vehicle routing problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 6, pp. 5867–5879, 2024.

TABLE II: Results obtained by the algorithms in comparison on the E-X benchmark suite. Values in bold indicate the best value for the problem instance in comparison. An asterisk ($*$) indicates the best-known values in the literature

Instance Name	Index	ILS	TS	SIGALNS	SA	VNS	GA	BACO	SAEA	IIRF
	min	**384.67**	**384.67**	**384.67**	**384.67**	**384.67**	**384.67**	**384.67**	**384.67**	385.39
E-n22-k4	mean	385.69	**384.67**	390.29	**384.67**	**384.67**	**384.67**	**384.67**	**384.67**	385.39
	std	2.11	0	1.22	0	0	0	0	0	0
	min	590.35	**571.94**	**571.94**	**571.94**	**571.94**	**571.94**	**571.94**	**571.94**	**571.94***
E-n23-k3	mean	592.05	**571.94**	**571.94**	**571.94**	**571.94**	**571.94**	**571.94**	**571.94**	572.97
	std	1.04	0	0	0	0	0	0	0	0.59
	min	**509.47**	**509.47**	**509.47**	**509.47**	**509.47**	**509.47**	**509.47**	**509.47**	**509.47***
E-n30-k3	mean	**509.47**	**509.47**	**509.47**	**509.47**	**509.47**	**509.47**	**509.47**	**509.47**	510.20
	std	0	0	0	0	0	0	0	0	0.92
	min	840.57	844.25	859.25	840.57	**840.14**	844.25	840.57	844.25	841.08
E-n33-k4	mean	844.07	845.62	960.08	854.07	**840.43**	844.62	842.30	844.33	841.78
	std	7.78	0.92	0.74	12.8	1.18	0.92	1.42	0.33	0.90
	min	**529.90**	**529.90**	537.72	533.66	**529.90**	**529.90**	**529.90**	**529.90**	530.10
E-n51-k5	mean	539.03	542.08	540.01	533.66	543.26	542.08	**529.90**	**529.90**	530.84
	std	6.92	8.57	4.63	0	3.52	8.57	0	0	0.67
	min	694.64	694.64	724.24	701.03	**692.64**	697.27	**692.64**	698.20	692.98
E-n76-k7	mean	704.24	704.24	762.09	712.17	697.89	717.30	**692.85**	698.20	697.01
	std	7.15	7.15	13.23	5.78	3.09	9.58	4.58	3.78	5.38
	min	841.02	841.02	891.77	845.84	839.29	852.69	840.25	851.96	**835.65**
E-n101-k8	mean	851.62	851.62	904.62	852.48	853.34	872.69	845.95	852.87	**843.03**
	std	6.93	6.93	4.64	3.44	4.73	9.58	4.58	3.78	3.71
	min	22310.64	16058.29	16947.33	16610.37	16028.05	16488.60	15901.23	16345.36	**15891.05**
X-n143-k7	mean	25119.80	16318.57	17239.63	17188.90	16459.31	16911.50	16031.46	16805.17	**15912.48**
	std	943.38	160.07	237.14	170.44	242.59	282.3	262.47	210	14.73
	min	16185.75	11323.56	12036.05	11404.44	11323.56	11762.07	11133.14	11639.52	**11064.82**
X-n214-k11	mean	16920.67	11537.58	12180.83	11680.35	11482.20	12007.06	11219.70	11910.48	**11143.49**
	std	496.57	72.55	125.64	116.47	76.14	156.69	46.25	139.69	32.25
	min	35605.66	27947.89	28846.14	27222.96	27064.88	28008.09	**26478.34**	28043.66	26585.32
X-n351-k40	mean	36740.77	28364.41	29069.14	27498.03	27217.77	28336.07	**26593.18**	28551.61	26669.52
	std	746.64	142.04	194.23	155.62	86.20	205.29	72.86	228.99	40.66
	min	42130.94	26511.28	26898.35	27222.96	25370.80	26048.21	24763.93	26203.2	**24683.15**
X-n458-k26	mean	43636.01	26726.69	27078.63	25809.47	25582.27	26345.12	24916.60	26663.81	**24793.61**
	std	1168.58	126.93	203.85	157.97	106.89	185.14	94.08	239.84	42.71
	min	70165.22	53102.46	56216.82	51929.24	52181.51	54189.62	53822.87	54791.01	**51738.86**
X-n573-k30	mean	72484.83	53507.46	56748.55	52793.66	52548.09	55327.62	54567.15	55956.01	**52007.04**
	std	1263.59	275.76	418.94	577.24	278.85	548.05	231.05	599.80	58.88
	min	100081.55	74409.65	76008.85	72549.9	71345.40	73925.56	70834.88	74061.84	**70109.07**
X-n685-k75	mean	104587.25	75087.58	76191.62	73124.98	71770.57	74509.03	71440.57	75267.34	**70282.32**
	std	2265.65	259.60	436.19	320.07	197.08	409.43	281.78	420.97	87.025
	min	109909.73	84298.43	85263.42	81392.78	81002.01	84034.73	80299.76	84405.66	**80167.01**
X-n749-k98	mean	110799.97	84860.28	86099.78	81848.13	81327.39	84759.79	80694.54	95298.06	**80345.47**
	std	814.88	287.26	351.02	275.26	176.19	376.10	223.91	385.54	87.24
	min	187872.81	168651.19	174100.00	165069.80	164290.00	170965.70	164720.80	172065.50	**160849.03**
X-n819-k171	mean	188662.32	169837.06	174268.20	165895.80	164926.40	172410.10	165565.80	173435.20	**161097.05**
	std	709.00	483.35	386.24	403.70	318.62	568.58	401.02	520.28	141.14
	min	392760.29	348733.86	363013.20	342796.90	341649.91	357391.60	342993.00	361482.90	**334719.21**
X-n916-k207	mean	393846.56	350822.41	364742.10	343533.90	342460.70	360269.90	345000.00	363368.30	**335118.84**
	std	1536.21	1177.08	508.50	556.98	510.66	229.19	905.72	775.68	161.84
	min	155657.35	79493.37	79085.16	78053.86	77476.36	78832.90	76297.09	78770.72	**75910.82**
X-n1001-k43	mean	162417.70	79928.29	80501.29	N/A	77920.52	79163.34	77434.33	79788.05	**76097.42**
	std	5879.00	265.91	431.72	306.27	234.73	N/A	719.87	386.61	91.35

TABLE III: Results obtained by the algorithms in comparison on the E-F-M-X benchmark suite. Values in bold indicate the best value for the problem instance in comparison. An asterisk ($*$) indicates the best-known values in the literature; two asterisks ($**$) indicate that IIRF advances the best-known values to date

Instance Name	Index	ILS	SIGALNS	MMAS	MA	BACO	CBACO	SAEA	IIRF
E-n29-k4-s7	min	397.65	**378.44**	382.81	**378.44**	**378.44**	381.48	**378.44**	**378.44***
	mean	402.45	388.30	383.47	379.10	383.01	381.48	379.10	**378.44**
E-n30-k4-s7	min	570.21	**569.54**	591.94	579.14	**569.54**	573.45	579.14	**569.54***
	mean	579.12	593.21	599.13	579.14	569.66	573.45	579.14	**569.54**
E-n35-k3-s5	min	520.08	**514.92**	529.20	**514.92**	**514.92**	516.67	**514.92**	515.49
	mean	531.01	537.38	534.91	**515.44**	515.87	516.67	**515.44**	515.82
E-n37-k4-s4	min	849.23	**847.03**	860.19	849.73	853.07	853.27	847.53	**845.72***
	mean	865.87	856.83	864.65	850.26	869.36	854.38	848.40	**848.03**
E-n60-k5-s9	min	579.41	547.95	570.70	528.77	528.77	528.77	528.77	**528.47***
	mean	592.09	569.75	590.30	543.26	**528.77**	**528.77**	540.92	529.74
E-n89-k7-s13	min	743.46	720.84	760.09	716.84	694.93	699.48	702.50	**690.87***
	mean	751.90	762.12	775.11	722.42	696.91	700.61	715.24	**692.69**
E-n112-k8-s11	min	890.91	868.40	914.89	581.70	**836.34**	842.87	844.57	836.89
	mean	902.50	918.07	944.66	859.44	841.50	849.87	858.98	**840.35**
F-n49-k4-s4	min	729.23	729.97	785.99	732.57	**727.75**	741.64	**727.75**	729.99
	mean	731.65	740.94	803.37	734.38	**727.75**	747.64	729.72	735.94
F-n80-k4-s8	min	250.54	250.85	266.44	251.68	245.31	247.71	249.97	**243.57**
	mean	264.89	264.52	269.69	253.17	245.93	248.34	251.96	**245.79**
F-n140-k5-s5	min	1275.84	1221.58	1221.40	1216.85	1179.05	1239.26	1178.50	**1175.24**
	mean	1297.61	1311.87	1240.70	1224.27	1188.21	1240.66	1204.37	**1181.27**
M-n110-k10-s9	min	832.46	855.73	891.36	833.81	**828.56**	860.87	832.98	829.48
	mean	851.64	887.70	908.03	834.79	**828.56**	861.74	834.50	830.61
M-n126-k7-s5	min	1054.35	1245.87	1103.84	1065.30	**1047.92**	1051.13	1053.14	1052.89
	mean	1120.56	1296.05	1145.46	1078.12	**1048.87**	1053.06	1060.29	1055.62
M-n163-k12-s12	min	1111.26	1082.87	1154.11	1095.58	1099.85	1077.24	1075.90	**1038.80**
	mean	1168.54	1215.30	1173.45	1120.30	1111.99	1086.67	1103.81	**1040.72**
M-n212-k16-s12	min	1450.46	1582.12	1461.86	1378.72	1413.06	1395.26	1364.81	**1307.91****
	mean	1512.54	1642.99	1479.17	1421.45	1432.45	1409.90	1401.68	**1326.38**
X-n147-k7-s4	min	16745.21	18265.28	17130.12	16649.96	16188.07	16499.82	16301.22	**15734.26****
	mean	17021.69	18717.33	17731.86	16710.26	16283.78	16769.140	16690.92	**15789.50**
X-n221-k11-s7	min	11814.67	13014.22	12655.57	11639.52	11865.85	11826.33	11512.44	**10985.35****
	mean	12021.45	13967.04	12793.00	11910.48	11991.78	11925.61	11849.86	**11070.60**
X-n360-k40-s9	min	28059.65	32921.25	28468.35	28043.66	28292.44	30033.89	27671.10	**26406.57**
	mean	29098.12	33988.51	28653.36	28551.61	28535.70	30192.19	28137.46	**26575.76**
X-n469-k26-s10	min	26188.54	34993.12	27718.93	26217.78	26657.54	29429.31	25991.88	**24650.91**
	mean	27264.54	34993.12	29734.97	26408.36	26944.20	35789.02	26417.12	**24735.05**
X-n577-k30-s4	min	58201.34	70166.19	58562.37	55534.16	56195.59	57871.59	54258.33	**51820.90**
	mean	59031.12	78787.85	59318.53	55854.48	56753.74	62478.94	55376.62	**52035.79**
X-n698-k75-s13	min	79899.74	89644.95	76541.45	74155.14	77083.95	80961.17	73525.74	**69785.58**
	mean	84512.62	91956.95	79555.01	74189.36	77257.15	86403.62	74347.42	**69918.70**
X-n759-k98-s10	min	85307.21	105776.65	88381.21	84405.66	86000.18	92069.56	82912.57	**79642.62**
	mean	92258.96	106442.64	88813.80	85298.06	88639.34	100992.53	83961.80	**79892.02**
X-n830-k171-s11	min	194601.52	194253.53	175073.36	172194.24	174724.62	182606.15	171216.70	**160732.05****
	mean	202359.05	195390.69	177119.16	173098.51	175100.89	193336.80	172488.23	**161104.12**
X-n920-k207-s4	min	384246.52	435672.12	351626.85	359458.47	362066.32	377822.88	357424.03	**333886.77**
	mean	395142.94	444429.72	359255.54	359947.22	364811.17	389200.04	359201.67	**334327.64**
X-n1006-k43-s5	min	79873.51	138231.67	91675.47	78917.10	82770.13	83201.63	77741.93	**75578.90**
	mean	80823.43	146730.92	93589.04	79099.70	82801.27	93954.54	78609.26	**75787.61**