Dynamic Electric Vehicle Routing Problem using Population-Based Ant Colony Optimization

Michalis Mavrovouniotis
ERATOSTHENES Centre of Excellence
Cyprus University of Technology
Limassol, Cyprus
michalis.mavrovouniotis@eratosthenes.org.cy

Changhe Li
School of Artificial Intelligence
Anhui University of Science and Technology
Hefei, China
changhe.lw@gmail.com

Charalambos Chrysostomou
ERATOSTHENES Centre of Excellence
Cyprus University of Technology
Limassol, Cyprus
charalambos.chrysostomou@eratosthenes.org.cy

Maria N. Anastasiadou
ERATOSTHENES Centre of Excellence
Cyprus University of Technology
Limassol, Cyprus
maria.anastasiadou@eratosthenes.org.cy

Shengxiang Yang
School of Computer Science
and Informatics
De Montfort University
Leicester, UK
syang@dmu.ac.uk

***Abstract*—The population-based ant colony optimization (P-ACO) algorithm has been proven effective in addressing the dynamic electric vehicle routing problem (DEVRP). In the DEVRP the travel time is subject to dynamic changes representing real-world traffic conditions. P-ACO maintains an archive of solutions that are used to update the pheromone trails. By default, the archive is updated in first in, first out fashion. In this study, we introduce a new update strategy based on the similarities of the solutions stored in the archive to maintain the diversity when updating the pheromone trails. The experimental results on different DEVRP test cases demonstrate the effectiveness of the proposed strategy in comparison with other existing ones.**

***Index Terms*—Ant colony optimization, population-list, electric vehicles, dynamic optimization.**

I. Introduction

With the widespread use of electric vehicles (EVs), the electric vehicle routing problem (EVRP) has received considerable attention in the last few years [1]. Nevertheless, the utilization of EVs by logistic companies imposes significant challenges such as shorter driving range due to the battery capacity, and longer charging times than refueling times for conventional vehicles. The main difference of the EVRP with the classical vehicle routing problem (VRP) is the additional energy consumption constraint of EVs and their need to potentially visit a charging station when serving customers. Considering that the classical VRP is an $\mathcal{NP}$-hard problem, it indicates that the EVRP is even more complex to solve [2]–[4].

A variety of EVRP applications have been addressed using exact [5]–[7] and metaheuristic methods [2], [8]–[10] with the vast majority considering static environments. However, in real-world logistic systems dynamic factors occur, and thus, EVRPs with dynamic environments must be considered. In [11] a multi-agent reinforcement learning approach is applied to a dynamic EVRP (DEVRP) in which the customer requests appear dynamically. For the same DEVRP a memetic algorithm with an adaptive local search and random immigrant scheme strategy is proposed [12]. In [13] an adaptive memetic algorithm is designed to address a DEVRP in which the demand of existing customers change dynamically. Similarly, a robust optimization model is developed for a DEVRP in which the demands are uncertain [14].

More recently, a population-based ant colony optimization (P-ACO) has been found effective in the DEVRP with traffic factors in comparison with other peer ACO algorithms [15]. This is because P-ACO has been especially designed to address dynamic optimization problems [16], [17]. In particular, P-ACO maintains a population-list with previously optimized solutions that are reused when a dynamic change occurs to enhance the re-optimization process. Assuming that the dynamic changes are small to medium, the pheromone trails generated by the archived solutions of the previous environment will be useful to the new environment.

The way in which the population-list is updated has a significant impact on the performance of P-ACO [18], [19]. Several update strategies have been investigated in static optimization problems, including the age-based, quality-based and the elitist-based strategy. The commonly used age-based strategy has been used to update the archive of P-ACO applied on DEVRPs with traffic factors [15]. In this study, the performance of P-ACO is further investigated with two existing aforementioned strategies, as well as with a newly introduced

This work is partially supported by the SmartPLAIGO: Advancing Public Health services through AI driven GIS optimization, funded by the European Union Recovery and Resilience Facility of the NextGenerationEU instrument through the Research and Innovation Foundation under the Grant Agreement No CODEVELOP-AG-SH-HE/0823/0112, and from the 'EXCELSIOR': ERATOSTHENES: Excellence Research Centre for Earth Surveillance and Space-Based Monitoring of the Environment H2020 Widespread Teaming project (www.excelsior2020.eu). The 'EXCELSIOR' project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 857510, from the Government of the Republic of Cyprus through the Directorate General for the European Programmes, Coordination and Development and the Cyprus University of Technology.

similarity-based strategy which aims to increase diversity that is essential when addressing dynamic optimization problems.

The rest of the paper is organized as follows. Section II defines the electric vehicle routing problem and describes how dynamic test cases are generated. Section III describes the P-ACO framework together with the investigated update strategies. The experimental results are given in Section IV. Section V provides the concluding remarks.

II. Electric Vehicle Routing Problem

A. Mathematical Formulation

The EVRP is modeled using a complete weighted graph $G = (N, A)$, where $N = \{0\} \cup I \cup F'$ is a set of nodes and $A = \{(i,j) \mid i,j \in N, i \neq j\}$ is a set of arcs linking these nodes [6], [20]. Each arc (i,j) is associated with a non-negative value d_{ij} representing the distance between nodes i and j. The central depot is denoted by node 0, whereas the set $I \subset N$ denotes n customers, with each customer $i \in I$ having an associated positive value δ_i indicating their delivery demand. The set $F' \subset N$ represents β_i node copies of each charging station $i \in F$ (i.e., $|F'| = \sum_{i \in F} \beta_i$) to accommodate multiple visits, if necessary, for EVs to each charging station $i \in F'$ [21]. The upper bound on the number of node copies for each charging station is denoted as $\beta_i = 2|I|$, reflecting the worst case scenario where one EV is required for each customer [22]. Each EV is characterized by a cargo load capacity, u_i ($0 \leq u_i \leq C$), and a battery charge level y_i ($0 \leq y_i \leq Q$), where C and Q represent the maximal cargo and the maximal battery charge level of the EV, respectively. The values u_i and y_i signify the cargo load and battery charge level of an EV upon arrival at node i, respectively. Particularly, for each traveled arc (i,j) an EV consumes hd_{ij} of the remaining battery charge level, where h is the energy consumption rate of the EVs.

The mathematical model for the EVRP is defined as follows:

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

s.t.

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

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

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

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

$$u_j \geq u_i - \delta_j x_{ij} - C(1 - x_{ij}), \forall i \in N, \forall j \in N, i \neq j, \tag{6}$$

$$0 \leq u_i \leq C, \forall i \in N, \tag{7}$$

$$u_0 = C, \tag{8}$$

$$y_j \leq y_i - hd_{ij} x_{ij} + Q(1 - x_{ij}), \forall i \in I, \forall j \in N, i \neq j, \tag{9}$$

$$y_j \leq Q - hd_{ij} x_{ij}, \forall i \in F', \forall j \in N, i \neq j, \tag{10}$$

$$0 \leq y_i \leq Q, \; \forall i \in N, \tag{11}$$

$$y_0 = Q, \tag{12}$$

$$x_{ij} \in \{0,1\}, \; \forall i \in N, \; \forall j \in N, i \neq j, \tag{13}$$

where Eq. (1) defines the objective function to minimize the total distance of the routes, constraint (2) ensures each customer is visited exactly once, constraint (3) handles multiple visits to recharging stations, constraint (4) establishes the flow conservation, constraints (5), (6) and (7) ensure delivery demand fulfillment, constraint (8) ensures that EVs start with full cargo load, constraints (9), (10) and (11) maintain a non-negative battery charge level, constraint (12) ensures that EVs start fully charged, and (13) is the binary decision variable for arc traversal.

B. Generating Dynamic Test Cases

The DEVRP with traffic factors is used to generate dynamic test cases in this work aiming to simulate the impact that traffic congestion has on the EVs' energy range [15]. Each generated DEVRP test case is characterized by the magnitude of change and the frequency of change[1].

For the DEVRP a factor that increases or decreases the value of arcs connecting nodes i and j is assigned every environmental period T, denoted as $d_{ij}(T)$, where $T = \lceil t/f \rceil$, f is the frequency of a dynamic change, and t is the algorithmic iteration count. The value of arc (i,j) changes as follows:

$$d_{ij}(T+1) = \begin{cases} d_{ij}(0) + \mathcal{R}_{ij}, & \text{if arc}(i,j) \in A_{rnd}(T), \\ d_{ij}(T), & \text{otherwise}, \end{cases} \tag{14}$$

where $\mathcal{R}_{ij}$ is a random number drawn from a normal distribution with a zero mean and standard deviation proportional to the initial value of the arc $d_{ij}(0)$, $A_{rnd}(T) \subset A$ is a set of randomly selected arcs with size $\lceil mn(n-1) \rceil$ in which their values are subject to change at environmental period index T, n is the size of the problem, and $m \in (0,1)$ is the magnitude of change.

This dynamic alteration may impact the feasibility of the generated solution. Specifically, an increase in travel time between two nodes due to traffic congestion may violate the energy constraints in Eq. (9), Eq. (10), and Eq. (11). Conversely, a decrease in traffic congestion may affect the solution's quality, as planned visits to charging stations may become unnecessary.

III. Population-Based ACO

A. Constructing Solutions

A colony of ω ants begin their journey from the central depot $\{0\}$, and concurrently select nodes from the customer set $I \subseteq N$, to build ω EVRP solutions. The selection of the next customer is based on two values associated with each arc (i,j): (1) the pheromone trails accumulated so far during the optimization process; and (2) problem-specific heuristic information. The probability with which ant k, currently at

[1] Available for download at: https://github.com/Mavrovouniotis/DynamicEVRP

Algorithm 1 Construct EVRP Solution

$step^k \leftarrow 0$
$T^k \leftarrow$ insert depot $\{0\}$ at position $step^k$
while ant k has not selected all customers **do**
  $step^k \leftarrow step^k + 1$
  $i \leftarrow$ find last node from T^k at position $step^k - 1$
  if $\mathcal{N}_i^k$ is not $\emptyset$ **then**
    $j \leftarrow$ select next customer from $\mathcal{N}_i^k$ using Eq. (15)
    $T^k \leftarrow$ insert customer j at position $step^k$
  else if $\mathcal{N}_i^k$ is $\emptyset$ && capacity constraint is violated **then**
    $T^k \leftarrow$ insert depot $\{0\}$ at position $step^k$
  else if $\mathcal{N}_i^k$ is $\emptyset$ && energy constraints are violated **then**
    $q \leftarrow$ select closest station from i
    T^k insert station q at position $step^k$
  end if
end while
$T^k \leftarrow$ insert depot $\{0\}$ at position $step^k + 1$

node i (either the depot, customer or a charging station), chooses the next node is defined as follows:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, & \text{if } j \in \mathcal{N}_i^k, \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

where τ_{ij} is the existing pheromone trail value, $\eta_{ij} = 1/d_{ij}(T)$ is the heuristic value at environmental period T, α and β determine the relative influence of τ_{ij} and η_{ij}, respectively, and $\mathcal{N}_i^k$ is the set of nodes from the customer set I for the k–th ant adjacent to node i that satisfy the following conditions: (a) customer j must not be visited; (b) the delivery demand associated with the customer j must not exceed the vehicle capacity when added to the current EV route; (c) the required energy to travel from customer i to the customer j must not exceed the battery capacity when added to the current EV route; and (d) customer j must have at least one charging station or the depot within the current energy range of the EV. It is worth noting that the possibility of insufficient energy to travel to any energy recharging station or back to the central depot is eliminated due to this energy range criteria [2].

To build a feasible EVRP solution a structured procedure is followed. Whenever ant k has no feasible customer choice, that is $|\mathcal{N}_i^k| = 0$, but there are unselected customers from the I set, it is translated that the EV must either return back to the depot because the vehicle capacity is violated or it must visit the closest energy recharging station. In this way, the depot and charging station nodes are selected according to the occasion and not probabilistically as the customers [20]. In case $|\mathcal{N}_i^k| = 0$ and all customers of set I are selected, it indicates that the EVRP solution is completed. The overall solution construction is described in Algorithm 1.

B. Updating Pheromone Trails

A population-list P_t is maintained and stores the iteration-best ant in every iteration t [18]. Specifically, whenever the iteration-best ant enters P_t a positive pheromone update is performed as follows:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau, \quad \forall\ (i,j) \in T^{ib}, \tag{16}$$

where T^{ib} is the iteration-best EVRP solution, $\Delta\tau = (\tau_{max} - \tau_0)/K$ is the constant amount of pheromone it deposits, $K = |P_t|$ is the size of the population-list, τ_{max} and τ_0 are the maximum and initial pheromone trail values, respectively. Note that τ_0 is also the minimum pheromone trail value.

The size of the population-list P_t is limited, and thus, the iteration-best ant at iteration t must replace an existing ant stored in the P_t when iteration $t > K$. Specifically, whenever a replacement occurs a negative pheromone update is performed as follows:

$$\tau_{ij} \leftarrow \tau_{ij} - \Delta\tau, \quad \forall\ (i,j) \in T^{r}, \tag{17}$$

where T^r is the EVRP solution of the ant being replace, $\Delta\tau = (\tau_{max} - \tau_0)/K$ is the constant amount of pheromone removed from its trails.

It is worth noting that the P-ACO algorithm does not utilize pheromone evaporation as conventional ACO algorithms [23], [24], which reduces the computation time required to update the pheromone from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

C. Updating the Population-List

Four update strategies are investigated in this work when the population-list P_t exceeds the specific limit K described in the following subsections. It is worth mentioning that the updating of the archive P_t corresponds to a global pheromone update in P-ACO.

1) Age-Based Strategy: The existing age-based strategy works as a first-in-first-out queue [18]. The eldest solution in the population-list is replaced by the iteration-best ant. This update strategy is denoted as `Age` in Section IV.

2) Quality-Based Strategy: In the existing quality-based strategy the quality of the worst solution in the population-list is compared with the solution generated by the iteration-best ant [18]. The latter solution replaces the former one only if its solution quality is better, otherwise the population-list remains unchanged. This update strategy is denoted as `Quality` in Section IV.

3) Elitist-Based Strategy: The existing elitist-based strategy keeps track of the best-so-far solution generated by the ants so that the population-list always contains that "elite" solution [18]. The iteration-best ant is compared with the elite solution stored in the population-list and if its solution quality is better then it replaces the elite solution; otherwise the iteration-best ant replaces the eldest solution of the population-list (excluding the elite solution). The pheromone trails of the elite solution always receive a higher amount of pheromone than the rest of the solutions in the population list. This update strategy is denoted as `Elitist` in Section IV.

4) Similarity-Based Strategy: In the proposed similarity-based strategy the quality of the most similar solution in the population-list is compared with the solution generated by the iteration-best ant. If the solution quality of the iteration-best is better; then it replaces the existing solution in the population-list. The similarity between two solutions is defined by the number of common arcs they have as follows:

$$S = 1 - \frac{dist(T^{ib}, T^{r})}{z}, \tag{18}$$

where $dist(T^{ib}, T^{r})$ is the total number of common arcs between solutions T^{ib} and T^{r}, and $z = \frac{|T^{ib}|+|T^{r}|}{2}$ is the average number of *steps* (in Algorithm 1) of the two solutions. Note that the number of *steps* may differ due to the number of visits in charging stations. This update strategy is denoted as `Similarity` in Section IV.

D. Reacting to Dynamic Changes

ACO algorithms are well-known for their robustness and have proven their effectiveness in dynamic optimization problems [25], [26]. Specifically, when a dynamic change occurs some ACO's pheromone trails from the previous environment become outdated, as they are associated with the previous optimum, whereas some other ACO's pheromone trails are common with the newly generate optimum [17], [27]. Additionally, in the case of the described DEVRP, the solutions archived in the population-list of P-ACO may become infeasible due to increased travel time that may violate energy constraints.

A simple way to address both the outdated pheromone trails and the infeasible solutions when using P-ACO, is to remove all the solutions stored in P_t when a dynamic change occurs. The effect on the pheromone trails is to uniformly set back to τ_0 and start the optimization process from scratch. This strategy is denoted as `Restart` in Section IV. The `Restart` strategy may be useful when the changing environments are not correlated (e.g., severe dynamic changes).

Differently, the pheromone trails of the previous environment will be most probably useful when the changing environments are correlated (e.g., small to medium dynamic changes), some previous pheromone trails are likely to remain useful in the new environment. Therefore, a more efficient approach is to use pheromone trails corresponding to the solutions stored in the population-list [16]. In the case of DEVRP, when a dynamic change occurs the solutions stored in the population-list may violate the energy constraints. In response, these solutions are repaired to adhere to energy constraints by adding energy stations where needed [15], accompanied by updates to the associated pheromone trails of the affected nodes.

IV. Experimental Study

A. Experimental Design

1) Dynamic Test Environments: Three stationary EVRP benchmark instances (`X-n143-k7-s4`, `X-n351-k40-s35`, and `X-n573-k39-s6`) are obtained from *2020 Congress on Evolutionary Computation*[2] competition for the Electric Vehicle Routing Problem [20]. In these instances, the values after n, k, and s in the instance name identify the number of customers, the minimum number of EVs a solution can have, and the number of charging stations, respectively. The magnitude of change m is set to slightly (i.e., $m = 0.05$ and $m = 0.1$), medium (i.e., $m = 0.25$ and $m = 0.5$), and severely (i.e., $m = 0.75$) changing environments. The frequency of change f is set to $25n$ iterations (scalable for each problem instance, i.e., $f = 3575$ for `X-n143-k7-s4`, $f = 8775$ for `X-n351-k40-s35`, and $f = 14325$ for `X-n573-k39-s6`) to allow sufficient time for ACO algorithms react in dynamic changes. Initially, all algorithms are executed for $100n$ iterations without any dynamic change allowing the algorithms to converge before the first dynamic change. Five test cases are generated from each static benchmark instance. Ten environmental changes are allowed for each dynamic test case and 30 independent runs are executed with different random seeds.

2) Parameter Tuning: The parameter settings for all P-ACO algorithms (i.e., `Restart`, `Age`, `Quality`, `Elitist`, and `Similarity`) in dynamic environments are set following the optimized parameters used in [26]. The colony size is set to $\omega = 25$ ants, the decision rule parameters are fixed at $\alpha = 1$ and $\beta = 5$, the initial pheromone trail value is set to $\tau_0 = 1/(n-1)$, the population-list size is set to $K = 3$, and the maximum pheromone trail value is to $\tau_{max} = 1$.

3) Performance Evaluation: The modified *offline performance* [28] is adopted as the performance metric, which is defined as follows:

$$\bar{P}_{offline} = \frac{1}{E}\sum_{t=1}^{E} C^{*}, \tag{19}$$

where E is the number of observations taken on every iteration, and C^{*} is the best solution quality value since the last dynamic change.

To investigate the behaviour of the update strategies in P-ACO, the diversity of the solutions stored in the population-list is also recorded as follows:

$$\bar{P}_{diversity} = \frac{1}{K(K-1)}\sum_{p=1}^{K}\sum_{q\neq p}^{K} S, \tag{20}$$

where K is the population-list size and S is the common arcs metric defined in Eq. (18).

B. Experimental Analysis and Discussion

Table I presents the mean results of $\bar{P}_{offline}$ (averaged over 30 runs) obtained by the P-ACO algorithms with different update strategies for the generated DEVRP test cases. The corresponding statistical results are presented in Table II, where *Kruskal–Wallis* tests were applied followed by posthoc paired comparisons using pairwise *Mann-Whitney* tests with

[2] Available for download at: https://github.com/Mavrovouniotis/e-cvrp_benchmark_instances

TABLE I: Mean results for offline performance of P-ACO algorithms in different DEVRP test cases.

P-ACO, $m \Rightarrow$	0.05	0.1	0.25	0.5	0.75
			X-n143-k7-s4		
Restart	17897.57	17140.28	17072.81	16885.41	16954.35
Age	17676.88	16986.09	16895.32	16763.24	16897.15
Quality	17889.00	17214.03	17118.47	16947.68	17105.71
Elitist	17940.35	17109.54	17133.22	16941.15	17161.98
Similarity	17613.72	16954.36	16864.06	16808.15	17012.33
			X-n351-k40-s35		
Restart	28615.01	28368.37	27824.96	28372.34	28265.79
Age	28461.78	28347.44	27778.59	28364.14	28255.41
Quality	28570.56	28384.07	27882.98	28191.21	28045.42
Elitist	28422.14	28263.18	27854.47	28343.26	28228.86
Similarity	28284.75	28070.35	27665.27	27754.91	27821.21
			X-n573-k30-s6		
Restart	54406.74	54718.72	53932.54	53640.49	53053.23
Age	54415.44	54690.12	53929.67	53632.27	53009.77
Quality	53995.75	53697.57	53120.54	52781.50	52418.03
Elitist	53914.28	54335.99	53692.99	53470.71	52942.28
Similarity	53299.51	52824.03	52342.23	52157.66	51704.00

Bonferroni correction. The results are shown as "+", "−", and "∼" when the first algorithm is significantly better than the second one, when the second algorithm is significantly better than the first one, and when the two algorithm are not significantly different, respectively. Moreover, the dynamic behaviour of P-ACO algorithms in terms of offline performance against the environmental changes is presented in Fig. 1. Finally, the population-list diversity for the first 1000 iterations after the first environmental change is presented in Fig. 2. From the experimental results, several observation can be made by comparing the behaviour of the investigated P-ACOs.

First, the proposed `Similarity` update strategy performs significantly better than the other competing strategies in all DEVRP test cases; see the comparisons in Table II. This is because the diversity of solutions stored in the population-list is maintained at higher levels during the optimization process which can be observed in Fig. 2. Therefore, P-ACO with `Similarity` update strategy utilizes more than one promising solution to begin the adaptation process when a dynamic change occurs.

Second, the commonly used `Age` update strategy performs significantly better than the existing `Quality` and `Elitist` update strategies on the small DVERP test cases, i.e., `X-n143-k7-s4`; see the comparisons in Table II. As the problem size increases the `Quality` and `Elitist` update strategies perform significantly better than the `Age` update strategy in most DEVRP tests cases, i.e., `X-n351-k40-s35` and `X-n573-k39-s6`. The `Quality` and `Elitist` update strategies compared to the `Age` place a stronger emphasis on the exploitation of the best quality solutions found so far rather than the iteration-best solutions. Therefore, it can contribute faster to the offline performance of P-ACO as it can be observed in most environmental changes in Fig. 1(b) and Fig. 1(c). However, there is a higher to get stuck in a local optima as it can be observed for `Quality` in Fig. 2 where the solutions stored in the population-list become identical.

Finally, the `Restart` update strategy is significantly worst

TABLE II: Statistical test results of comparing the offline performance of P-ACO algorithms in different DEVRP test cases.

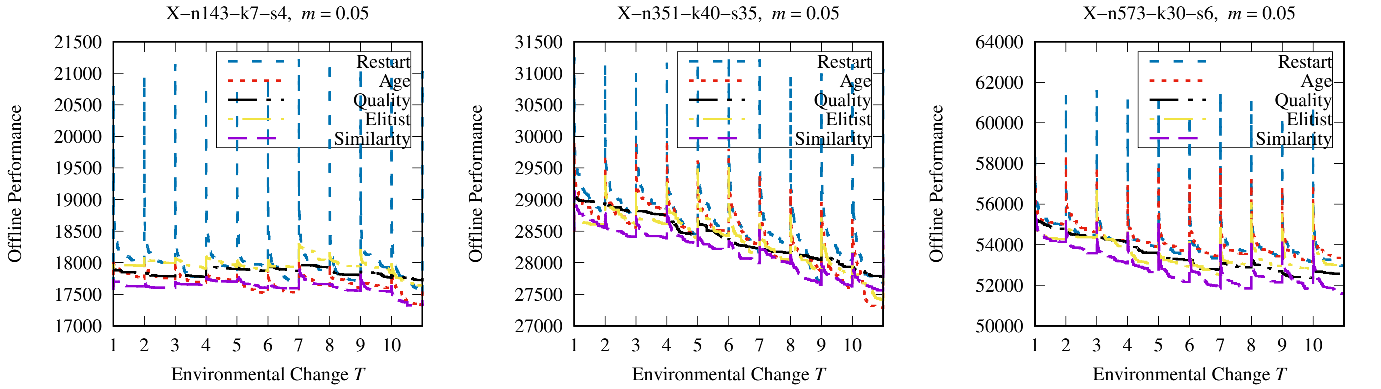
Comparisons, $m \Rightarrow$	0.05	0.1	0.25	0.5	0.75
			X-n143-k7-s4		
Age vs Restart	+	+	+	+	+
Age vs Quality	+	+	+	+	+
Age vs Elitist	+	+	+	+	+
Quality vs Restart	∼	∼	∼	∼	−
Quality vs Elitist	∼	∼	∼	∼	∼
Similarity vs Restart	+	+	+	+	∼
Similarity vs Age	+	∼	∼	∼	∼
Similarity vs Quality	+	+	+	+	+
Similarity vs Elitist	+	+	+	+	+
			X-n351-k40-s35		
Age vs Restart	+	∼	∼	∼	∼
Age vs Quality	∼	∼	∼	−	−
Age vs Elitist	−	−	+	∼	∼
Quality vs Restart	∼	∼	∼	+	+
Quality vs Elitist	∼	∼	∼	+	+
Similarity vs Restart	+	+	+	+	+
Similarity vs Age	+	+	+	+	+
Similarity vs Quality	+	+	+	+	+
Similarity vs Elitist	+	+	+	+	+
			X-n573-k30-s6		
Age vs Restart	∼	∼	∼	∼	∼
Age vs Quality	−	−	−	−	−
Age vs Elitist	−	−	−	−	∼
Quality vs Restart	+	+	+	+	+
Quality vs Elitist	+	+	+	+	+
Similarity vs Restart	+	+	+	+	+
Similarity vs Age	+	+	+	+	+
Similarity vs Quality	+	+	+	+	+
Similarity vs Elitist	+	+	+	+	+

than the remaining update strategies in almost all DEVRP cases, except in the `X-n573-k39-s6` which is competitive with the `Age` update strategy. It can be observed from Fig. 1 that due to the large spikes in offline performance of `Restart` on every environmental change the adaptation is much slower. A similar effect can be observed for the `Age` update strategy in Fig. 1(c).

V. Conclusions

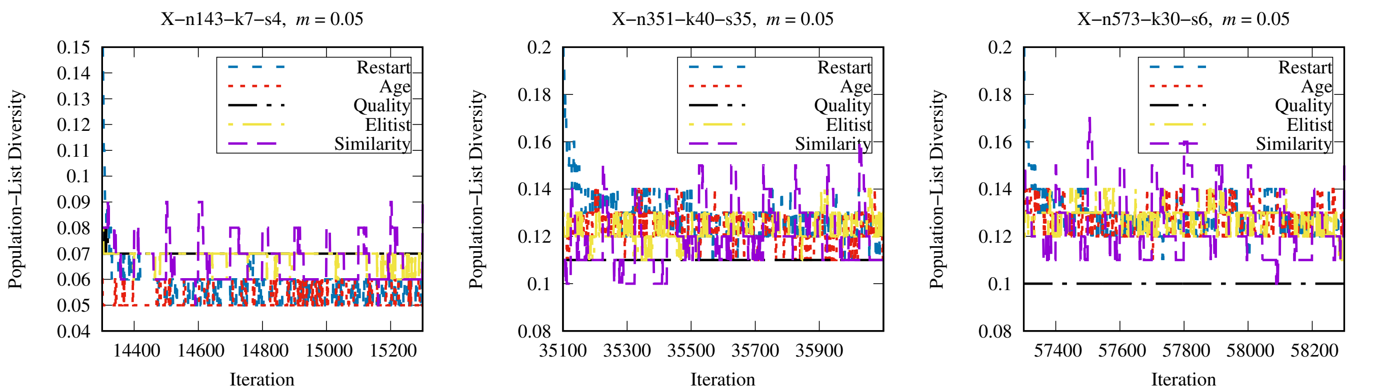
In this work, the effect of population-list update strategy has on the P-ACO performance is investigated. The DEVRP with traffic factors is utilized to systematically generate dynamic test cases. The default age-based strategy is compared against the quality-based and elitist-based strategies (commonly used in static environments) and the newly proposed similarity-based strategy.

From our experiments several key observation emerge. First, adapting to the dynamic changes by reusing the population-list solutions rather than restarting P-ACO performs significantly better in all DEVRPs cases. Second, the proposed similarity-based update strategy performs better since it maintains the diversity in the population-list. Third, the age-based update strategy is more explorative than the quality-based and elitist-based update strategies since it is focused on the iteration-best ants rather than the best-so-far ants. Therefore, the age-based update requires more time to adapt in the environmental changes.



(a) (b) (c)

Fig. 1: Offline performance of P-ACOs with different population-list update strategies on DEVRP test cases with $m = 0.05$.



(a) (b) (c)

Fig. 2: Population-list diversity of P-ACOs after the first environmental change with different update strategies on DEVRP test cases with $m = 0.05$.

For future research, it would be beneficial to develop more sophisticated update strategies for the population-list of P-ACO to address dynamic environments.

References

[1] M. Thymianis, A. Tzanetos, E. Osaba, G. Dounias, and J. Del Ser, "Electric vehicle routing problem: Literature review, instances and results with a novel ant colony optimization method," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE Press, 2022, p. 1–8.

[2] M. Mavrovouniotis, C. Li, G. Ellinas, and M. Polycarpou, "Parallel ant colony optimization for the electric vehicle routing problem," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2019, pp. 1660–1667.

[3] M. Mavrovouniotis and S. Yang, *Ant colony optimization for dynamic combinatorial optimization problems*, ser. Control, Robotics and Sensors. Institution of Engineering and Technology, 2018, pp. 121–142.

[4] M. Mavrovouniotis, G. Ellinas, C. Li, and M. Polycarpou, "A multiple ant colony system for the electric vehicle routing problem with time windows," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2022, pp. 796–803.

[5] J. Lin, W. Zhou, and O. Wolfson, "Electric vehicle routing problem," *Transportation Research Procedia*, vol. 12, pp. 508–521, 2016, tenth International Conference on City Logistics 17-19 June 2015, Tenerife, Spain.

[6] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle-routing problem with time windows and recharging stations," *Transportation Science*, vol. 48, pp. 500–520, 2014.

[7] X. Zuo, Y. Xiao, M. You, I. Kaku, and Y. Xu, "A new formulation of the electric vehicle routing problem with time windows considering concave nonlinear charging function," *Journal of Cleaner Production*, vol. 236, p. 117687, 2019.

[8] V. Q. Hien, T. C. Dao, and H. T. T. Binh, "A greedy search based evolutionary algorithm for electric vehicle routing problem," *Applied Intelligence*, vol. 53, no. 3, pp. 2908–2922, 2023.

[9] Y.-H. Jia, Y. Mei, and M. Zhang, "Confidence-based ant colony optimization for capacitated electric vehicle routing problem with comparison of different encoding schemes," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2022.

[10] Y.-H. Jia, Y. Mei, and M. Zhang, "A bilevel ant colony optimization algorithm for capacitated electric vehicle routing problem," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.

[11] T. Zhou, M. L. Kris, D. Creighton, and C. Wu, "Gmix: Graph-based spatial–temporal multi-agent reinforcement learning for dynamic electric vehicle dispatching system," *Transportation Research Part C: Emerging Technologies*, vol. 144, p. 103886, 2022.

[12] J. Dong, H. Wang, and S. Zhang, "Dynamic electric vehicle routing problem considering mid-route recharging and new demand arrival using an improved memetic algorithm," *Sustainable Energy Technologies and Assessments*, vol. 58, p. 103366, 2023.

[13] N. Wang, Y. Sun, and H. Wang, "An adaptive memetic algorithm for

dynamic electric vehicle routing problem with time-varying demands," *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 6635749, 2021.

[14] X. Liu, D. Wang, Y. Yin, and T. Cheng, "Robust optimization for the electric vehicle pickup and delivery problem with time windows and uncertain demands," *Computers & Operations Research*, vol. 151, p. 106119, 2023.

[15] M. N. Anastasiadou, M. Mavrovouniotis, and D. Hadjimitsis, "Ant colony optimization for the dynamic electric vehicle routing problem," in *Parallel Problem Solving from Nature – PPSN XVIII*, M. Affenzeller, S. M. Winkler, A. V. Kononova, H. Trautmann, T. Tušar, P. Machado, and T. Bäck, Eds. Cham: Springer Nature Switzerland, 2024, pp. 68–84.

[16] M. Guntsch and M. Middendorf, "Applying population based ACO to dynamic optimization problems," in *Ant Algorithms*, ser. Lecture Notes in Computer Science, M. Dorigo, G. Di Caro, and M. Sampels, Eds., vol. 2463. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 111–122.

[17] M. Mavrovouniotis and S. Yang, "Ant colony optimization with immigrants schemes for the dynamic vehicle routing problem," in *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science, C. Di Chio, A. Agapitos, S. Cagnoni, C. Cotta, F. de Vega, G. Di Caro, R. Drechsler, A. Ekárt, A. Esparcia-Alcázar, M. Farooq, W. Langdon, J. Merelo-Guervós, M. Preuss, H. Richter, S. Silva, A. Simões, G. Squillero, E. Tarantino, A. Tettamanzi, J. Togelius, N. Urquhart, A. Uyar, and G. Yannakakis, Eds. Springer Berlin Heidelberg, 2012, vol. 7248, pp. 519–528.

[18] M. Guntsch and M. Middendorf, "A population based approach for ACO," in *EvoWorkshops 2002: Applications of Evolutionary Computing*, ser. LNCS, vol. 2279. Springer-Verlag, 2002, pp. 72–81.

[19] S. Oliveira, M. S. Hussin, A. Roli, M. Dorigo, and T. Stützle, "Analysis of the population-based ant colony optimization algorithm for the tsp and the qap," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 1734–1741.

[20] 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.

[21] S. Erdoğan and E. Miller-Hooks, "A green vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 100–114, 2012.

[22] 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.

[23] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, Feb 1996.

[24] M. Mavrovouniotis and S. Yang, "Empirical study on the effect of population size on $\mathcal{MAX}$-$\mathcal{MIN}$ ant system in dynamic environments," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC 2016)*, 2016, p. 853–860.

[25] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press, 1999.

[26] M. Mavrovouniotis, S. Yang, M. Van, C. Li, and M. Polycarpou, "Ant colony optimization algorithms for dynamic optimization: A case study of the dynamic travelling salesperson problem [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 15, no. 1, pp. 52–63, 2020.

[27] D. Angus and T. Hendtlass, "Ant colony optimisation applied to a dynamically changing problem," in *Developments in Applied Artificial Intelligence*, ser. Lecture Notes in Computer Science, T. Hendtlass and M. Ali, Eds., vol. 2358. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 618–627.

[28] J. Branke and H. Schmeck, "Designing evolutionary algorithms for dynamic optimization problems," in *Advances in Evolutionary Computing*, ser. Natural Computing Series, A. Ghosh and S. Tsutsui, Eds. Springer Berlin Heidelberg, 2003, pp. 239–262.