IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 November 2023, accepted 21 November 2023, date of publication 24 November 2023, date of current version 29 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3336810

RESEARCH ARTICLE

A Hybrid K-Means and Particle Swarm Optimization Technique for Solving the Rechargeable E-Scooters Problem

MAHMOUD MASOUD
Department of Information Systems and Operations Management, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia
Center for Smart Mobility and Logistics, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia

e-mail: Mahmoud.masoud@kfupm.edu.sa

This work was supported by the King Fahd University of Petroleum and Minerals during the academic year 1444AH/2023AD.

ABSTRACT E-scooters are gaining popularity for short-distance travel, but their recharging presents challenges. To reduce their downtime, we propose a Hybrid K-Means/Particle Swarm Optimisation (PSO) approach, optimizing charging routes using machine learning and meta-heuristics. The research in this paper attempts to determine if a combination of a meta-heuristic such as PSO and a machine learning algorithm for clustering such as K-Means, would be effective at solving the vehicle routing problem for e-scooters. We compared this method with other algorithms and found that Tabu Search excelled in over 95% of tests. While Hybrid K-Means/PSO led in only approximately 52% of scenarios, it was also the only one to provide an output that surpassed Tabu Search in one of the scenarios. The core difference in efficiency is due to traditional meta-heuristic methods providing routes that while optimal, may also travel from locations relatively far from each other, while Hybrid K-Means/PSO will provide routes between locations that are clustered and in local groups. This results in Hybrid K-Means/PSO being slightly less efficient but may be more practical for charging personnel as they can operate in designated areas close to each other rather than a more optimal route with nodes further apart. This research underscores the effectiveness of Tabu Search and the potential of our Hybrid K-Means/PSO approach for optimizing e-scooter charging routes.

INDEX TERMS E-scooter rechargeable, hybrid optimization k-means/particle swarm, tabu search, guided local search, simulated annealing.

I. INTRODUCTION

With the popularization of e-scooters comes a new way of transportation to complete that last-mile journey. This popular form of transportation, however, has its own new set of problems to tackle in order to become better and more efficient. Some such issues include e-scooters needing to be recharged and their generally bulky size can make it difficult for freelance chargers to bring multiple home to recharge them [17], [18]. In order to remedy this, rechargeable batteries are being gradually implemented to become the norm. This means that the operators that charge these e-scooters can bring dozens of batteries on their person at a time and go on a route where they simply need to replace the battery of the scooter, decreasing the amount of time and effort needed to charge any one scooter [1], [2], [16]. This process, however, can be made to be more efficient as it is possible to optimize the route and minimise the amount of distance the operators have to travel which will lower fuel costs and time investment [11].

This problem can be overtly described as a similar problem to the capacitated vehicle routing problem, which in itself is derivative of the traveling salesman problem. The generally idea is that a traveling salesman has to go from location to location selling their goods, the salesman wishes to minimize the overall distance travelled as it will consume less time. The vehicle routing problem is essentially the same but with multiple salesman instead of one, where the capacitated vehicle routing problem includes adding a capacity to both the vehicle travelling around and the locations the vehicles arrive at. This means that the vehicles not only have to minimize the overall distance their routes take, but also supply the

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

necessary capacity of goods at each location based on how much they can carry.

This paper presents a new method using K-means and Particle Swarm Optimisation in order to solve this problem, while also comparing it against other methods. These methods include Guided Local Search, Simulated Annealing, Tabu Search, and Greedy Descent, all of which will be implemented using Google OR-Tools for convenience. The hybridization of PSO and K-Means clustering can be beneficial for companies in the decision-making process of deploying operators for managing scooters. While K-Means clustering is effective in determining the optimal number of operators and assigning them to specific areas [26], it does not consider the efficiency of travel distance. On the other hand, PSO excels at optimizing travel routes but may not provide an accurate allocation of operators across different areas. By combining these two algorithms, the hybrid approach leverages the strengths of both. K-Means clustering helps in determining the appropriate number of operators and assigning them to specific regions based on factors like demand or population density. This ensures efficient scooter management and availability in different areas. This hybridization allows companies to strike a balance between convenient operator deployment and efficient travel distance, resulting in an optimized scooter management system.

The field of e-scooter optimization is relatively new, and as such, there is a scarcity of literature exploring this specific domain. One paper utilizes a mixed-integer linear programming (MILP) model along with comparisons between an adapted college admission algorithm (ACA) and black hole optimizer (BHO) [17]. The solution presented in their paper attempts to provide optimal route assignments but will produce results that have operators overlapping in areas to produce the best route with the smallest distance. The method presented here attempts to create separate designated areas for operators and to the best of our knowledge is among the first to apply this technique. The core idea behind this decision is through the implementation of designated local areas based off of k-means, the allocation of operators will be more simple and effective, where PSO is used to determine the most effective routes in each area. This lack of research poses significant research gaps in understanding the optimal deployment and management of e-scooters. This paper contributes to the existing literature by addressing these research gaps through the application of different algorithms. By utilizing various optimization techniques such as Particle Swarm Optimization (PSO), K-Means clustering, and comparing them with algorithms like Tabu Search, Greedy Descent, Guided Local Search, and Simulated Annealing, the paper expands the knowledge base in e-scooter optimization. The application of these algorithms provides valuable insights into the effectiveness and efficiency of different approaches, paving the way for further advancements in optimizing e-scooter systems. In this paper we are proposing a new approach using K-means and Particle Swarm Optimisation in order to solve the vehicle routing problem for e-scooters. The e-scooters vehicle routing problem will also be solved using other methods such as Guided local Search, Simulated Annealing, Tabu Search and Greedy Descent in order to compare the effectiveness of our method against traditional meta-heuristics. These methods were chosen due to the fact that these meta-heuristics are fitting to the nature of the problem and are easy to implement with Google OR-Tools and can be compared against our method with little friction.

A. RECHARGEABLE E-SCHOOTERS

Public rechargeable e-scooters are a type of e-scooter that can be rented and used by anyone, similar to how public bikes are used in many cities. These scooters can be found in designated areas throughout a city and can be easily accessed using a smartphone app. Once a user has finished using the scooter, they can leave it at any designated drop-off location for the next person to use. Public rechargeable e-scooters have become a popular transportation option in many cities around the world due to their convenience and environmental benefits. Unfortunately, e-scooter sharing is a relatively new business, with many issues left unaddressed. One major issue to highlight is the environmental life cycle of an e-scooter, where although it produces no emissions during use, the charging of the e-scooter does. From the emergence of public e-scooters as a method of micro-transportation, they have been recharged by freelancers by taking the whole e-scooter and charging it in a different location and then bringing it back. This method is gradually being replaced by a new method where the e-scooters are fitted with a replaceable battery [17], [19] and can be charged on the street, no longer needing to collect them. With this emerges a new issue, and that is how will these replaceable batteries be delivered. A potential solution being explored that is also eco-friendly is using e-cargo bikes, currently these bikes have a capacity between 20-40 batteries and have a range of approximately 30km. As the environmental footprint of the recharging of e-scooters is quite significant, as it amounts to 43% of the total environmental impact of an e-scooter, reducing it as much as possible could make a substantially positive influence as a whole.

B. CAPACITATED VEHICLE ROUTING PROBLEM

The Vehicle Routing Problem (VRP) can be described as an extension of the Travelling Salesman Problem (TSP). Where given a list of locations, the objective is to minimize the total cost (i.e., the distance or travel time) needed to serve all the customers. Unlike TSP, VRP is defined as having multiple salesmen or vehicles which can travel around to any of the locations in any order as long as all locations are visited with the objective of shortening the overall combined distance of all vehicles. The Capacitated Vehicle Routing Problem (CVRP) is defined as a VRP with vehicles that have a limited carrying capacity and the need to pick up or deliver items at the given locations. As the problem being tackled in this paper involves recharging e-scooter batteries using a number of operators, it can be seen how these dilemmas can be equated with each other. One issue with determining an optimal solution to VRP is that it is classified as NP-hard, which means if the problem is too large, optimally solving it using mathematical programming may be difficult as it would simply take too much time to run. Thus, it is much more practical to use heuristics to solve this type of problem as the size and frequency of real world VRPs tend to be significant [5].

As VRPs generally relate to the service of a delivery company, a depot or location where the vehicles must start out from and return to is generally required to be included. In our case a depot would also be necessary as it is assumed there will be a location that the operators need to go to pick up the replaceable batteries. Normally, this problem would also model when it is impossible to satisfy all of the customer's demands, however, in this case it was assumed that the minimum number of operators needed to charge every scooter will be available as the exact number of scooters and load capacity of each operator is known and not variable [3], [8].

The problem of recharging e-scooters has been presented as a multi-depot periodic vehicle routing problem (MDPVRP) in [3]. The MDPVRP problem is the challenge of constructing a set of routes for each day of a specified p-day period for a homogenous fleet of vehicles of capacity Q. Day k routes must be accessed by mk vehicles located in the depot designated for day k. Each vehicle can only travel one route each day, and each route must begin and end at the same depot. Each e-scooter may demand visits on fi (say) different days over the time, and these visits may only occur in one of a limited number of day combinations. Also, the mixed-integer linear programming (MILP) is developed [17].

II. METHODOLOGY

The implementation of the separate methods for optimizing the routes for the capacitated vehicle routing problem described in this paper can be split into five major components.

1. Cleaning and extraction of data.
2. Implementing a machine learning method (K-means) on the data to create clusters of locations.
3. Running PSO on the produced clusters (Hybrid method) in step 2, to produce optimized routes.
4. Run the data through the Google OR-Tools API with the different methods provided (i.e., Guided Local Search, Simulated Annealing, etc...).
5. Compare and analyse the results

Figure 1 shows the main framework of the proposed methodology.

A. DATA EXTRACTION

The E-Scooter Trips 2019 Chicago Pilot dataset is a considerably large dataset containing 710839 e-scooter trips with information detailing the trip distance, time, location, and area. For our algorithm, only the location of the e-scooter at the end of the trip is required as that is presumably when a scooter may need charging. The overall data will

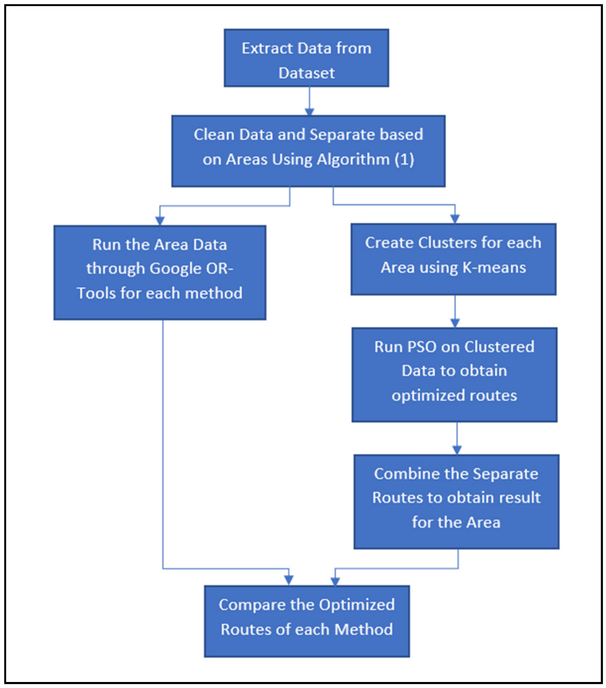

FIGURE 1. M methodology flowchart.

be cleaned through removing rows that contain NaN values in the relevant columns needed, of which include the End Community Area Number, End Centroid Latitude and End Centroid Longitude. In order to evaluate the effectiveness of the various route optimization methods over many different scenarios, the data is first split into its separate areas. This will produce multiple sub datasets that can be evaluated, such as a dataset specifically in the West Town area of Chicago. These separate datasets will produce results showing how effective each method is under different circumstances. Table 1 shows Relevant Columns Used for Implementation.

During extraction of the data, one quirk found within the dataset was the fact that there would often times be multiple scooters in the same location at the same time stamp. This caused some issues as the dataset only provided unique trip IDs and not unique scooter IDs, making it difficult to identify whether or not it was multiple scooters ending at that location or one scooter being used multiple times within the hour. In order to remedy this, we made the assumption that every unique trip would be treated as a unique scooter, but the maximum number of scooters at any one location would be limited to 15. The reason for this assumption was due to the fact that it is unreasonable to expect that no scooters will be parked next to each other and also unreasonable to expect there to be dozens if not hundreds of scooters parking in the same spot over the course of a day. The data was also in latitudinal and longitudinal coordinates which needed to be converted to X and Y coordinates. This was done using the geometric parameters of the World Geodetic System 1984 (WGS84) ellipsoid [9]. Cartesian coordinates can be calculated from geographical coordinates using the following equations.

$$X = (v + h) \cos(\phi) \cos(\lambda) \tag{1}$$

TABLE 1. Units for magnetic properties.

Column Name	Description
Trip ID	A unique code used for identifying each trip
End Time	Timestamp of when the trip ended, rounded to the nearest hour
End Community Area Number	A number identifying the Community Area where the trip ended
End Community Area Name	The Community Area name where the trip ended
End Centroid Latitude	The Latitude of the centre of the trip end census tract
End Centroid Longitude	The Longitude of the centre of the trip end census tract

$$Y = (v + h) \cos(\phi) \cos(\lambda) \tag{2}$$

where ϕ is latitude, λ is longitude, h is height, and v is the radius of the curvature in the prime vertical plane and is given as:

$$v = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \tag{3}$$

where e^2, the square of the first eccentricity is given by

$$e^2 = f(2 - f) \tag{4}$$

The geometric parameters of WGS84 are:
Semi-major axis $a = 6378137\ m$
Flattening $f = 1/298.257223563$
Semi-minor axis $b = a(1 - f) = 6356752.3142\ m$
The pseudo-code for the data extraction can be seen below:

Algorithm (1):

1. ***Load*** *data as* $Dataset_{all}$
2. ***Remove*** *unnecessary rows with NaNs as values for relevant columns in* $Dataset_{all}$ *as* $Dataset_{cleaned}$
3. ***For*** *'i' = number of community areas:*
   a. ***Extract*** *all rows and columns of community area 'i' in 'End Community Area Number' from* $Dataset_{cleaned}$ *as* $Dataset_{Area_i}$
   b. ***Find*** *timestamp* t_{max} *in 'End Time' from* $Dataset_{Area_i}$ *with maximum number of rows*
   c. ***Extract*** *all rows and columns of timestamp* t_{max} *in 'End Time' from* $Dataset_{Area_i}$ *as* $Dataset_{Area_i}$
   d. ***Extract*** *'End Centroid Latitude' and 'End Centroid Longitude' columns from* $Dataset_{Area_i}$ *as* $Dataset_{Area_i}$
   e. ***Convert*** $Dataset_{Area_i}$ *from 'End Centroid Latitude' and 'End Centroid Longitude' to 'X' and 'Y':*
      i. ***Initialise*** $Dataset_{AreaXY_i}$ *as empty*
      ii. ***Initialise*** *parameters of WGS84,* $a = 6378137, f = 1/298.257223563$, $b = a(1-f),\ e^2 = f(2-f),\ h = 0$
      iii. ***Convert*** $Dataset_{Area_i}$ *from degrees to radians using* $Dataset_{AreaRadians_i} = Dataset_{Area_i} \times \left(\frac{\pi}{180}\right)$
      iv. ***For*** *'j' = number of rows in* $Dataset_{AreaRadians_i}$:
         1. ***Calculate*** *'v' using Equation (3):* $v = \frac{a}{\sqrt{1-e^2 \sin^2 \phi}}$
         2. ***Calculate*** *'X' using Equation (1):* $X = (v+h)\cos(\phi)\cos(\lambda)$
         3. ***Calculate*** *'Y' using Equation (2):* $Y = (v+h)\cos(\phi)\cos(\lambda)$
         4. ***Append*** *['X', 'Y'] onto* $Dataset_{AreaXY_i}$
4. ***Output*** $Dataset_{AreaXY_i}$ *as the datasets to be used for evaluation in each algorithm*

B. CLUSTERING USING K-MEANS

K-means is a machine learning technique that is one of the most widely used clustering techniques in the world due to its simplicity and speed. The way k-means works, is through first arbitrarily choosing k number of initial centres, after which it seeks to minimise the average squared distance between points in the same cluster by moving the centroids through constant iterations until they no longer change. The end outcome is a k number of clusters each consisting of the closest datapoints surrounding the centroids. Constrained K-means is simply k-means given specific parameters constraining the centroids such that they meet the specified minimum and maximum group of nodes for each cluster [13], [20], [21].

With the data fully prepared for evaluation, the Area datasets can then be clustered using K-means clustering. Through finding the number of scooters in the area, it is possible to determine the number of clusters needed based on the number of batteries any single operator can carry. As the assumption made in this paper is that e-scooters will be charged using replaceable batteries, a single operator can carry a considerable number. While there is no consensus on battery size and weight, for the purpose of this methodology we will assume that any and all operators can carry 30 replaceable batteries as e-cargo bikes can carry approximately 20-40 [21]. The 'k' number of clusters will always be determined to be the minimum number of clusters required to charge every e-scooter, for example if there are 91 scooters, then k will equal 4. Once the area dataset has been clustered, each cluster can then be extracted to be used for evaluation using PSO.

The pseudo-code for the implementation of constrained K-means can be seen below:

1. ***Define*** *constraint parameters of the clusters:* $size_{min} = 10,\ size_{max} = 30$
2. ***Calculate*** *number of clusters needed using* $n_{clusters} = ceil\left(\frac{len\left(Dataset_{AreaXY_i}\right)}{size_{max}}\right)$, *where ceil is 'ceiling', i.e. round up, and* $len(Dataset_{AreaXY_i})$ *is the number of data points in the dataset*
3. ***Cluster*** *the data using the constrained k-means function through inputting the number of clusters, minimum size, maximum size and* $Dataset_{AreaXY_i}$.

C. PARTICLE SWARM OPTIMISATION

Particle Swarm Optimization (PSO) is an optimization algorithm used for primarily minimising the cost of an objective function. It optimizes a problem through scattering particles (solutions) into a search space and then iteratively moving those particles in order to find the global minimum. These particles move in the direction of their respective local minimums and the global minimum at velocities dependent on how far away from either point they are. The local minimum is the best solution that the particle has obtained so far and is called pbest, while the global minimum is the best solution that the whole group of particles has obtained so far and can be denoted as gbest, [12], [14]. Once the two best values are found, the position and velocity of the particles are updated by Equations 5 and 6.

$$v_i = wv_i + c_1r_1\left(pbest_i - x_i\right) + c_2r_2\left(gbest - x_i\right) \quad (5)$$

$$x_i = x_i + v_i \quad (6)$$

where v_i is the velocity of the i^{th} particle, w, c_1, and c_2 are constants that define the weight of each variable, r_1 and r_2 are random variables in the range 0 to 1 and x_i is the solution of the i^{th} particle. $pbest_i$, as stated previously, is the personal best solution of i^{th} particle while $gbest$ is the best solution found from all particles [10]. As PSO is being used to solve a problem similar to the travelling salesman problem, the implementation of the algorithm will have to be modified for a discrete array of routes and distances. An existing method to solve this particular type of problem is called the Swap Operator (SO) and Swap Sequence (SS) [4], [12], [15], [23].

The swap operator can be defined as followed, consider a normal solution sequence of TSP with $S = (a_i)\,, i = 1 \ldots n$.

The Swap Operator $SO\left(i_1, i_2\right)$ denotes the indices of two locations in the route and is used to swap location a_{i_1} and location a_{i_2} in the solution S. This new solution can then be defined as

$$S' = S + SO\left(i_1, i_2\right) \quad (7)$$

To take as an example, if a solution $S = (2, 1, 4, 5, 3)$, and the swap operator is $SO\left(3, 4\right)$, then the new solution S' is

$$\begin{aligned} S' = S + SO(3, 4) &= (2, 1, 4, 5, 3) + (3, 4) \\ &= (2, 1, 5, 4, 3) \end{aligned} \quad (8)$$

where $+$ is defined as applying the swap operator in order to swap the position of the location nodes in the route.

The Swap Sequence SS is essentially a list containing one or more swap operators.

$$SS = \left(SO_1, SO_2, SO_3 \ldots, SO_n\right) \quad (9)$$

$SO_1, SO_2, \ldots SO_n$ are swap operators, and the order of the SOs in the swap sequence is important [10]. A swap sequence being applied to a route means all of the swap operators of that SS are applied in order which can be described by the following formula:

$$\begin{aligned} S' = S + SS &= S + \left(SO_1, SO_2, SO_3 \ldots, SO_n\right) \\ &= \left(\ldots\left(\left(S + SO_1\right) + SO_2\right) + \ldots + SO_n\right) \end{aligned} \quad (10)$$

Using this method of interaction for the particles, the original formula will then have to be changed in order to make it applicable, where the formula is now

$$v_i = v_i \oplus r_1\left(pbest_i - x_i\right) \oplus r_2\left(gbest - x_i\right) \quad (11)$$

where $\left(pbest_i - x_i\right)$ and $\left(gbest - x_i\right)$ are now SSs and $\oplus$ is defined as merging two swap sequences into a new swap sequence. This in execution is essentially when a merged SS is applied to a solution, SS_1 is applied and then the second SS_2 is applied afterwards.

Once the data had been clustered, Particle Swarm Optimization can be applied to determine the optimal route for each cluster. As the PSO method being implemented uses swap operators and is used for evaluating the traveling salesman problem, the algorithm needs to be separately applied on each cluster. This will produce multiple routes originating from the depot, after which the results can be combined to produce the final routes and distance. This depot needs to be independently created and inserted into the dataset as an assumption to where the e-scooter company may set up a location to store batteries. For this, the depot is created using the mean of the latitudes and longitudes of the area, where we are assuming that the depot will be in the approximate centre of the area. As we are predominantly comparing the optimisation capabilities of the algorithms, the same will be done for the case where all the locations in Chicago are compared and will not have multiple depots introduced.

The pseudo-code for the PSO algorithm used can be seen below:

Algorithm (2):

1. ***Initialize*** *particles with randomized routes.*
   Calculate the distance travelled for each route and treat them as $pbest_i$*. Set gbest as the* $pbest_i$ *with the smallest distance.*
2. ***For*** *each particle:*
   a. ***Calculate*** *the Swap Sequences through finding the difference between* $pbest_i$ *and* x_i *and determine velocity using Equation (11):*
      $v_i = v_i \oplus r_1\left(pbest_i - x_i\right) \oplus r_2\left(gbest - x_i\right)$
   b. ***Calculate*** *new route* x_i *using Equation (6):*
      $x_i = x_i + v_i$
   c. ***Update*** $pbest_i$ *is the new route has a smaller distance than the previous route*
3. ***Update*** *gbest if a route with a smaller distance than the previous gbest exists*
4. ***If*** *stopping criteria (for e.g. 1000 iterations) is reached, then take gbest as the solution and output. Else, return back to step 2 and loop.*

The trade-off between exploration and exploitation capabilities in PSO is effectively handled through the hybrid approach by incorporating the strengths of K-Means clustering. PSO inherently possesses strong exploration capabilities [25], allowing particles to explore the search space to discover potentially better solutions. However, this exploration might lead to a slower convergence towards the optimal solution. In the hybrid approach, the K-Means clustering component helps strike a balance between exploration and exploitation. By using K-Means clustering, the algorithm identifies the optimal number of clusters or regions in the problem space. This process ensures that particles in PSO are distributed across different clusters, enabling both exploration and exploitation to take place effectively. Each cluster represents a specific region of interest, and particles within that cluster explore and exploit solutions within their localized area. This allows for focused search and exploitation of the best solutions within each cluster. The hybrid approach strikes a balance between exploring diverse regions of the problem space (through cluster assignment) and exploiting the best solutions within each region (through PSO's particle interactions). This synergy enhances the algorithm's overall performance and increases the likelihood of finding high-quality solutions efficiently.

D. IMPLEMENTING THE OTHER METHODS

Using Google's OR-Tools API, it is possible to quickly set up code that will run optimization methods for the capacitated vehicle routing problem. Such methods include Guided Local Search, Simulated Annealing, Tabu Search and Greedy Descent. Unlike PSO with K-means, these methods do not need the area data to be clustered and then run on each cluster. These methods were run directly on the area data after specifying the specific number of operators and the location of the depot.

1) SIMULATED ANNEALING

Simulated Annealing is a heuristic optimization algorithm that uses randomness to find the global minimum or maximum of a function. It is based on the concept of annealing in metallurgy, where a material is heated and then slowly cooled to increase its hardness and reduce defects. In simulated annealing, the solution to a problem is represented as a point in a search space, and the algorithm uses random moves to explore different points in this space [22].

At each step, the algorithm considers a new solution and decides whether to accept it based on its quality compared to the current solution. If the new solution is better, it is always accepted. If it is worse, it is sometimes accepted with a probability that depends on how much worse it is, and a parameter called the temperature. The temperature is gradually decreased over time, which causes the algorithm to become more selective and converge on the optimal solution. The key to the success of simulated annealing is the temperature schedule, which determines how quickly the temperature decreases and how selective the algorithm becomes over time. A well-designed temperature schedule can help the algorithm avoid getting stuck in local minima and find the global minimum.

Pseudo-code for the general formulation of the Simulated Annealing algorithm can be seen below:

Algorithm (3):

1. ***Initialize*** *solution (i.e. in this case a set of routes determined by using the First Solution Strategy: "PATH_CHEAPEST_ARC" from Google OR-Tools). PATH_CHEAPEST_ARC starts from a route "start" node and iteratively connects to the next node that is the shortest distance away from it until it returns to the "start" node.*
2. ***Initialize*** *'temperature' variable*
3. ***While*** *the stopping condition has not been met (for e.g. 1000 iterations) and the temperature is above a minimum threshold:*
   a. ***Generate*** *a set of potential routes by performing local search from the current route*
      i. ***Select*** *a random route from the set of routes*
      ii. ***Select*** *a random e-scooter from the route*
      iii. ***Generate*** *a set of possible new routes by removing the selected e-scooter from the selected route and inserting it into each of the other routes*
      iv. ***Evaluate*** *each of the new routes based on distance and load capacity*
      v. ***Select*** *the new route with the smallest distance travelled that still satisfies the load constraint*
   b. ***If*** *the new route is an improvement over the previous route, update the set of routes with the new route*
   c. ***Else if*** *the new route is not an improvement, accept the new route based on a probability equal to* $e^{-\frac{\Delta}{T}}$, *where* Δ *is the difference in distance between routes, and* T *is the temperature*
   d. ***Decrease*** *the temperature based on a pre-defined cooling schedule*
   e. ***If*** *stopping condition has not been met return to 3a and loop.*
4. ***If*** *the stopping condition has been met, output the current set of routes as the optimal solution*

2) GUIDED LOCAL SEARCH

Guided local search is a metaheuristic optimization algorithm that combines elements of local search and constraint satisfaction techniques. It is called a metaheuristic because it is a higher-level algorithm that guides the search for a solution, rather than a specific solution algorithm for a particular problem. The algorithm works by first starting with an initial solution to a problem, and then iteratively improving the solution by making local changes. This process is guided by a set of constraints that must be satisfied, as well as a quality function that evaluates the goodness of a solution [5].

During each iteration, the algorithm first applies a local search procedure to the current solution to try and find a better one. This can involve making small changes to the solution, such as swapping two items in a list or adjusting the values of variables. The algorithm then evaluates the new solution using the quality function, and accepts it if it is an improvement over the current solution. If the new solution does not satisfy the constraints, the algorithm may use a repair procedure to modify the solution so that it satisfies the constraints. This process is repeated until the algorithm converges on a satisfactory solution.

Pseudo-code for the general formulation of the Guided Local Search algorithm can be seen below:

Algorithm (4):

1. ***Initialize*** *solution (i.e. in this case a set of routes determined by using the First Solution Strategy: "PATH_CHEAPEST_ARC" from Google OR-Tools). PATH_CHEAPEST_ARC starts from a route "start" node and iteratively connects to the next node that is the shortest distance away from it until it returns to the "start" node.*
2. ***While*** *the stopping condition has not been met (for e.g. 1000 iterations):*
    a. ***Generate*** *a set of potential routes by performing local search from the current route*
        i. ***Select*** *a random route from the set of routes*
        ii. ***Select*** *a random e-scooter from the route*
        iii. ***Generate*** *a set of possible new routes by removing the selected e-scooter from the selected route and inserting it into each of the other routes*
        iv. ***Evaluate*** *each of the new routes based on distance and load capacity*
        v. ***Select*** *the new route with the smallest distance travelled that still satisfies the load constraint*
    b. ***If*** *the new route is an improvement over the previous route, update the set of routes with the new route*
    c. ***Else if*** *the new route is not an improvement, apply a probability factor to determine whether to accept the new route anyway*
    d. ***If*** *stopping condition has not been met return to 2a and loop.*
3. ***If*** *the stopping condition has been met, output the current set of routes as the optimal solution*

3) TABU SEARCH

Tabu search is a metaheuristic optimization algorithm that is used to find good solutions to problems that are difficult to solve using traditional optimization techniques. The algorithm works by iteratively improving a solution to a problem by making small changes to the current solution, while avoiding making the same change multiple times. This is done by maintaining a list of "tabu" moves, which are changes that are not allowed to be made again in the current search.

The tabu search algorithm begins with an initial solution to the problem, and then iteratively makes changes to the solution in order to improve it. At each step, the algorithm evaluates the possible moves that can be made to the current solution, and selects the best move that is not tabu. If no such move can be found, the algorithm may make a move that is tabu, but with a reduced probability. The algorithm continues to make moves until it reaches a predefined stopping condition, such as a maximum number of iterations or a satisfactory level of solution quality [7], [11], [16], [24].

Pseudo-code for the general formulation of the Tabu Search algorithm can be seen below:

Algorithm (5):

1. ***Initialize*** *solution (i.e. in this case a set of routes determined by using the First Solution Strategy: "PATH_CHEAPEST_ARC" from Google OR-Tools). PATH_CHEAPEST_ARC starts from a route "start" node and iteratively connects to the next node that is the shortest distance away from it until it returns to the "start" node.*
2. ***Initialise*** *the Tabu List to be empty*
3. ***While*** *the stopping condition has not been met (for e.g. 1000 iterations):*
    a. ***Generate*** *a set of potential routes by making small changes to the current route (e.g. using 2-opt which is swapping the order of the locations of two e-scooters on a route)*
    b. ***Evaluate*** *each potential route to determine distance travelled*
    c. ***Choose*** *the best route with the smallest distance travelled that is currently not on the Tabu List*
    d. ***Update*** *the Tabu List by adding the current route onto it*
    e. ***If*** *stopping condition has not been met return to 3a and loop.*
4. ***If*** *the stopping condition has been met, output the current set of routes as the optimal solution*

III. NUMERICAL RESULTS

The following subsections implement the methodology for the route optimization algorithms, display the results and summaries the findings.

A. IMPLEMENTING THE OTHER METHODS

A flowchart diagram detailing the process for our hybrid PSO K-Means method can be seen in Figure 2. Where the extracted and processed data is clustered using constrained K-means and is then evaluated through PSO. Through using constrained K-Means, it is possible to create as many clusters as is needed to separate each cluster into no more than 30 scooters. The constraints for the clustering are a maximum of 30, a minimum of 10, with k equal to the number of scooters divided by 30 rounded up. These constraints

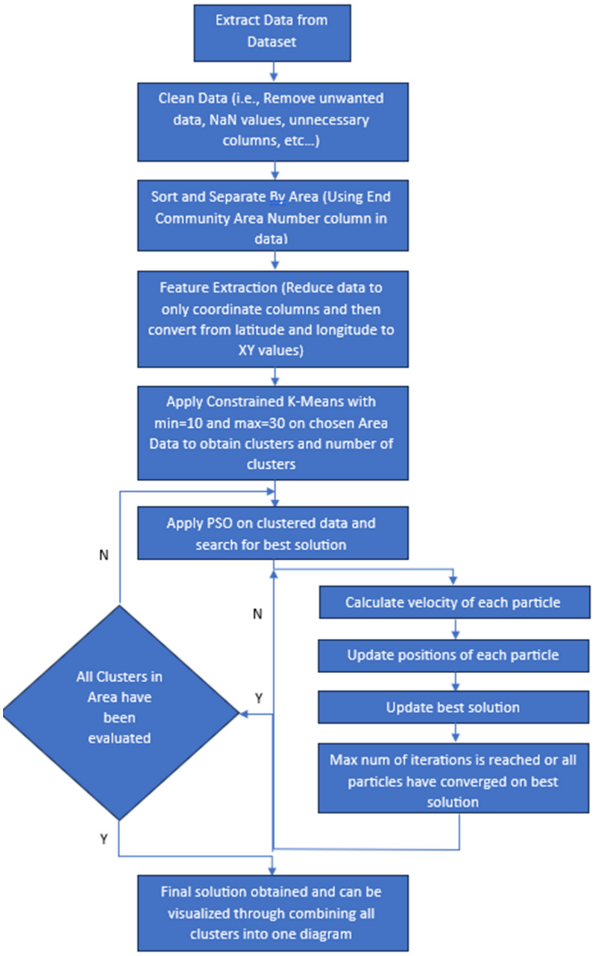

FIGURE 2. Hybrid PSO K-Means flowchart diagram.

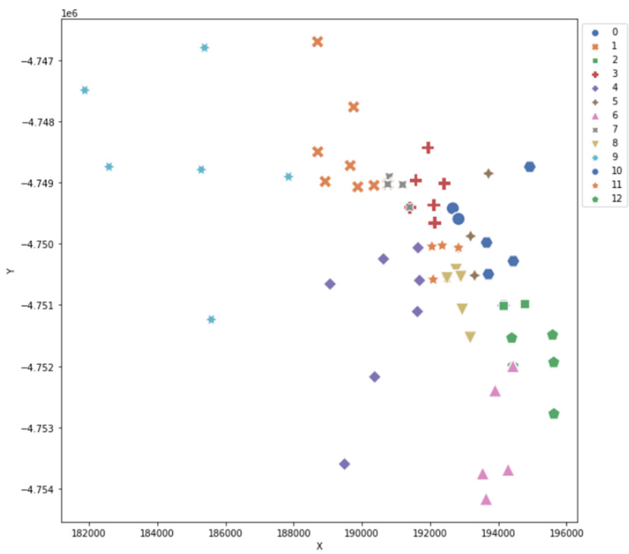

FIGURE 3. E-Scooters in Chicago clustered.

produce Figure 3 when applied on all of Chicago at a single timestamp.

With the locations of the scooters clustered, particle swarm optimization could be applied to solve what is essentially the travelling salesman problem. By applying PSO separately on each cluster, it will produce the optimized route from the depot and back, where an example of the result can be seen in Figure 4 below.

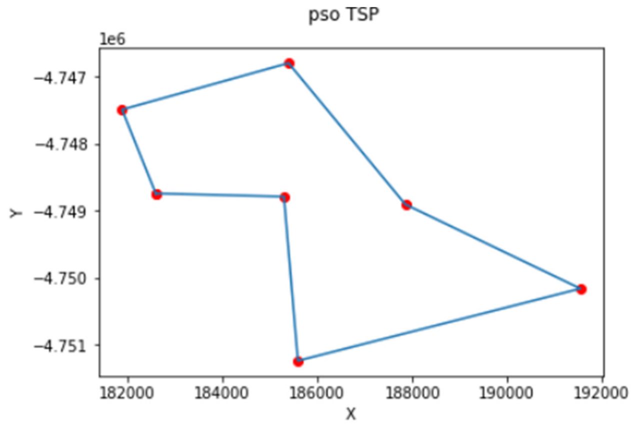

FIGURE 4. PSO with K-Means for cluster 1 of Chicago.

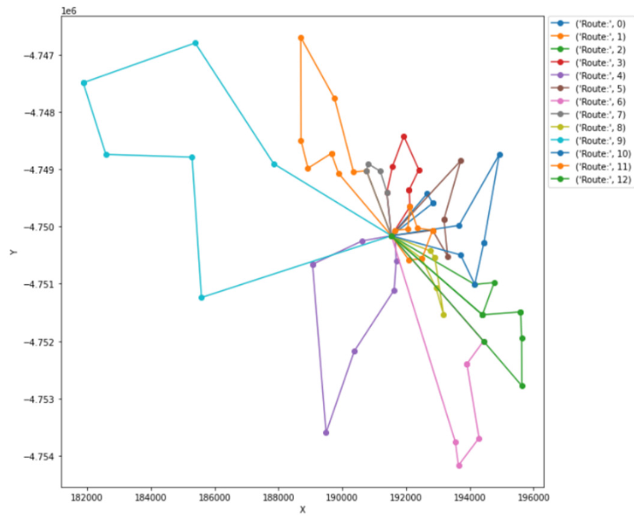

FIGURE 5. PSO with K-Means for Chicago.

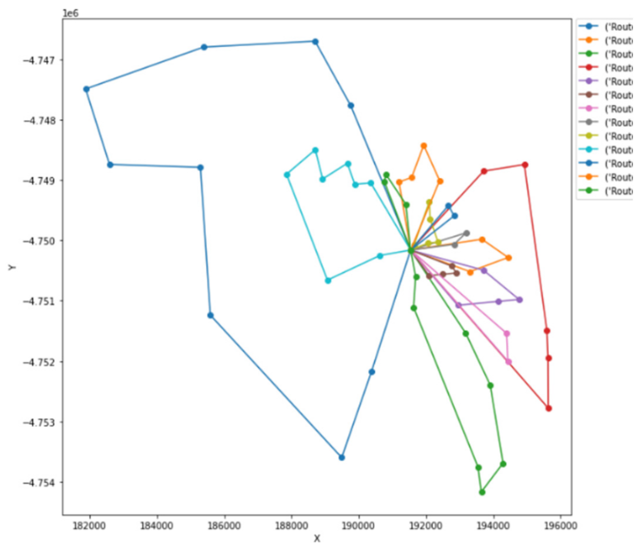

FIGURE 6. TS for Chicago.

TABLE 2. Comparison table of results for all methods.

Area	Load	No. of Locations	PSO+K-Means	Guided Local Search	SA	TS	Greedy Descent
7	31	5	5147	5147	5147	5147	5147
8	32	6	6882	6882	6882	6882	6882
9	84	8	11883 ↑ 1.05%	11759	12143 ↑3.27%	11759	12143 ↑3.27%
10	15	3	1326	1326	1326	1326	1326
11	53	11	10231	10231	12226 ↑19.5%	10231	12226 ↑19.5%
13	30	5	2185	2185	2185	2185	2185
14	44	8	4816 ↑ 0.417%	4796	4796	4796	4796
15	453	34	46874 ↑ 2.76%	45617 ↑0.009%	45873 ↑0.570%	45613	45873 ↑0.570%
16	32	8	7514 ↑ 0.697%	7462	7462	7462	7517 ↑0.737%
17	525	36	51501 ↑ 4.24%	49623 ↑0.443%	49702 ↑0.603%	49404	49702 ↑0.603%
18	364	26	42790 ↑ 4.75%	41576 ↑1.77%	42005 ↑2.82%	40851	42005 ↑2.82%
19	18	4	2175	2175	2175	2175	2175
20	22	3	1786	1786	1786	1786	1786
21	322	23	37201	37648 ↑1.20%	37904 ↑1.89%	37857 ↑1.76%	37904 ↑1.89%
22	40	4	4160	4160	4160	4160	4160
24	40	7	7563	7563	7563	7563	7563
25	40	6	8213 ↑ 1.99%	8053	8053	8053	8053
45	35	4	4695 ↑ 25.3%	3746	3746	3746	3746
67	65	6	12337 ↑ 1.80%	12119	12119	12119	12119
76	45	3	3910	3910	3910	3910	3910
All	377	58	106044 ↑ 6.17%	102433 ↑2.56%	111503 ↑11.6%	99878	111503 ↑11.6%
Mean % Increase from Best			↑2.34%	↑0.285%	↑1.92%	↑0.084%	↑1.95%

With all of the clusters routes solved using PSO, we can then combine them into a single figure to obtain a more encompassing visualisation of how the operators will charge the scooters in their respective routes. From Figure 5 below, it can be seen that there is one depot where all operators start and end at.

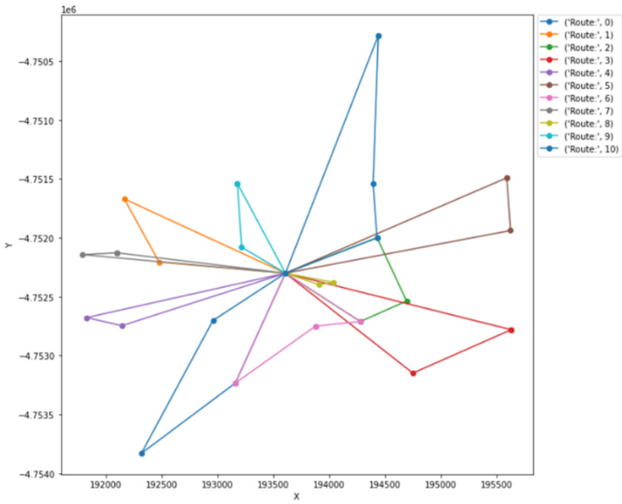

FIGURE 7. PSO with K-Means for area 21.

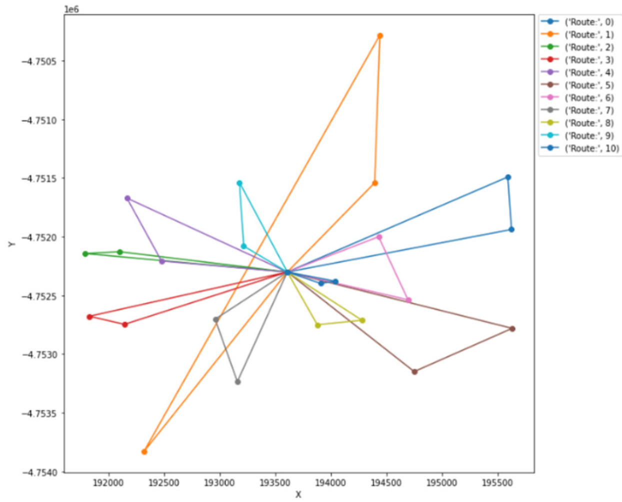

FIGURE 8. Guided local search for area 21.

The results obtained from Google OR-Tools do not need to be combined as they were not separated using clusters, as such the end result is directly output and can be seen below in Figure 6.

From comparing the two figures, it can be observed that PSO with K-means produces results where the locations are all in similar directions and very rarely overlap unless one of the location nodes has too many scooters for one operator to handle. Tabu search on the other hand will sometimes have routes that surround other routes but never have multiple operators arrive at the same destination.

As can be seen from the comparison Table 2 below, the highlighted values in green show which algorithm performed the best and produced the smallest distance for each respective area. Each value in the columns beneath the meta-heuristic methods is the best result out of 10 runs using each method for each area. The algorithm with the largest number of highlighted values is Tabu Search, with Guided Local Search being not too far behind. The method of utilising K-Means with Particle Swarm Optimization ultimately did not have ideal performance, with only very few scenarios where it produces a smaller cost than the other methods. This may be due to the fact that it is only capable of creating routes in localised areas as a result of the clustering and removes that step outside of the optimization. This can mean that sometimes the optimal solution may require a route to reach a further destination than in the local cluster before returning to the depot. In order to make sure all of the algorithms were compared under similar conditions; each algorithm was computed using 1000 iterations. Table 2 includes a percentage increase value for each of the non-best distances in the table, for example in row 3 PSO+K-Means is 1.05% (11883/11759) greater than Guided Local Search and TS. The mean % increase is the sum of all percentage increases divided by the number of cases (21) for each algorithm respectively.

When looking at one of the solutions where PSO with K-Means performs better than the other methods, it can be seen that in Figure 7 that the routes have all of their nodes occur in the same direction away from the depot. Whereas, in Figure 8 the second best result from GLS has routes that go in multiple directions away from the depot, resulting in extra distance travelled. This is most likely due to the fact that GLS never visits the same node twice, while PSO with K-Means does not have that restriction allowing routes like route 1 from Figure 8 to be avoided as other operators can also visit those locations. While the restriction that only one vehicle can visit one location makes sense in the context of the original vehicle routing problem, as customers would only want to be visited once by someone who will serve them fully. It is not as fundamental rule for our case as different people going to change the batteries of scooters in the same location will not inconvenience any customers.

IV. CONCLUSION

This paper detailed the development, analysis and comparison of a route optimisation algorithm using PSO and K-means against other similar methods such as Guided Local Search, Simulated Annealing, Tabu Search and Greedy Descent. The optimisation algorithms were applied onto an e-scooter trip dataset from the 2019 Chicago Pilot in order to determine their effectiveness as a method for optimising e-scooter battery charging routes. The results showed that although in certain cases it was possible for the PSO with K-means method to produce the best result, the majority of the time Tabu Search was the best algorithm for this scenario. Guided Local Search was the algorithm with the second highest number of best costs, with Simulated Annealing being slightly worse and Greedy Descent with the fourth highest number of best costs. One thing to take of note is that while PSO with K-means overall performed the weakest, it was the only algorithm to produce better results in some cases when compared to Tabu Search, whereas the other methods were almost always the same or generally worse.

The main contribution of this paper is the proposal of a novel hybrid approach that combines Particle Swarm

Optimisation (PSO) and K-Means clustering, leveraging their respective strengths to address the challenges in efficient scooter deployment and route optimisation. The integration of PSO ensures effective exploitation capabilities for route optimisation, while K-Means clustering enables optimal allocation of operators based on demand and population density. This hybridisation strikes a balance between convenient operator deployment and efficient travel distance, resulting in an optimised scooter management system. Additionally, the paper presents empirical results based on real-world data from the 2019 Chicago pilot program, highlighting the superior performance of Tabu Search algorithm in optimizing charging routes for e-scooters. The findings provide valuable insights for companies seeking to enhance the management of scooter charging operations.

REFERENCES

[1] M. H. Almannaa, H. I. Ashqar, M. Elhenawy, M. Masoud, A. Rakotonirainy, and H. Rakha, "A comparative analysis of e-scooter and e-bike usage patterns: Findings from the city of Austin, TX," *Int. J. Sustain. Transp.*, vol. 15, no. 7, pp. 571–579, May 2021.

[2] M. H. Almannaa, F. A. Alsahhaf, H. I. Ashqar, M. Elhenawy, M. Masoud, and A. Rakotonirainy, "Perception analysis of E-scooter riders and non-riders in Riyadh, Saudi Arabia: Survey outputs," *Sustainability*, vol. 13, no. 2, p. 863, Jan. 2021.

[3] A. Mingozzi, "The multi-depot periodic vehicle routing problem," in *Proc. Int. Symp. Abstraction, Reformulation, Approximation*, Berlin, Germany, 2005, pp. 347–350.

[4] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Comput. Ind. Eng.*, vol. 57, no. 4, pp. 1472–1483, Nov. 2009.

[5] C. Voudouris, E. P. K. Tsang, and A. Alsheddy, "Guided local search," in *Handbook of Metaheuristics*. Boston, MA, USA: Springer, 2003, pp. 185–218.

[6] Chicago, IL, USA. (Apr. 15, 2020). *E-Scooter Trips—2019 Pilot*. [Online]. Available: https://data.cityofchicago.org/Transportation/E-Scooter-Trips-2019-Pilot/2kfw-zvte

[7] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, Jan. 1986.

[8] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Oper. Res.*, vol. 12, no. 4, pp. 568–581, Aug. 1964.

[9] G. P. Gerdan and R. E. Deakin, "Transforming Cartesian coordinates X,Y,Z to geographical coordinates φ, λ, h," *Austral. Surveyor*, vol. 44, no. 1, pp. 55–63, 1999.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Perth, WA, Australia, 1995, pp. 1942–1948.

[11] J. Hollingsworth, B. Copeland, and J. X. Johnson, "Are e-scooters polluters? The environmental impacts of shared dockless electric scooters," *Environ. Res. Lett.*, vol. 14, no. 8, Aug. 2019, Art. no. 084031.

[12] K.-P. Wang, L. Huang, Z. Chun-Guang, and W. Pang, "Particle swarm optimization for traveling salesman problem," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Xi'an, China, 2003, pp. 477–480.

[13] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained K-means clustering with background knowledge," in *Proc. 18th Int. Conf. Mach. Learn.*, Williamstown, MA, USA, 2001, pp. 577–584.

[14] M. A. H. Akhand, S. Akter, S. S. Rahman, and M. M. H. Rahman, "Particle swarm optimization with partial search to solve traveling salesman problem," in *Proc. Int. Conf. Comput. Commun. Eng. (ICCCE)*, Kuala Lumpur, Malaysia, Jul. 2012, pp. 118–121.

[15] M. Jaberipour, E. Khorram, and B. Karimi, "Particle swarm algorithm for solving systems of nonlinear equations," *Comput. Math. with Appl.*, vol. 62, no. 2, pp. 566–576, Jul. 2011.

[16] M. Masoud, M. Elhenawy, M. H. Almannaa, S. Q. Liu, S. Glaser, and A. Rakotonirainy, "Optimal assignment of e-scooter to chargers," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 4204–4209.

[17] M. Masoud, M. Elhenawy, M. H. Almannaa, S. Q. Liu, S. Glaser, and A. Rakotonirainy, "Heuristic approaches to solve E-scooter assignment problem," *IEEE Access*, vol. 7, pp. 175093–175105, 2019.

[18] M. Masoud, M. Elhenawy, S. Q. Liu, M. Almannaa, S. Glaser, and W. Alhajyaseen, "A simulated annealing for optimizing assignment of E-scooters to freelance chargers," *Sustainability*, vol. 15, no. 3, p. 1869, Jan. 2023.

[19] N. Carey and P. Lienert, "E-scooters fall head over wheels for battery swapping," *Reuters*, p. 1, Mar. 2022.

[20] P. S. Bradley, K. P. Bennett, and A. Demiriz, "Constrained K-means clustering," *Microsoft Res.*, vol. 20, pp. 1–8, May 2000.

[21] S. Severengiz, S. Finke, N. Schelte, and N. Wendt, "Life cycle assessment on the mobility service E-scooter sharing," in *Proc. IEEE Eur. Technol. Eng. Manage. Summit (E-TEMS)*, Dortmund, Germany, Mar. 2020, pp. 1–6.

[22] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[23] Sapna and M. Kaur, "Particle swarm optimization to solve multiple traveling salesman problem," *Int. Res. J. Eng. Technol.*, vol. 4, no. 7, pp. 1179–1184, 2017.

[24] W. E. Featherstone, "An explanation of the geocentric datum of Australia and its effects upon future mapping," *Cartography*, vol. 23, no. 2, pp. 1–12, Dec. 1994.

[25] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *Proc. IEEE Congr. Evol. Comput., IEEE World Congr. Comput. Intell.*, Hong Kong, Jun. 2008, pp. 1128–1134.

[26] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, Aug. 2020, doi: 10.3390/electronics9081295.

MAHMOUD MASOUD received the Ph.D. degree in operations research and mathematical sciences from the School of Mathematical Sciences, Queensland University of Technology (QUT), Brisbane, Australia. He was a Research Associate with the Centre for Accident Research and Road Safety—Queensland (CARRS-Q), QUT. He is currently an Associate Professor with the King Fahd University of Petroleum and Minerals (KFUPM). He has extensive experience, as a Research Associate, in many industrial projects as a part of effective teamwork with the Centre for Tropical Crops and Bio-Commodities (CTCB), School of the Mathematical Sciences, and CARRS-Q, QUT. This team constructed industrial linkages with big industrial organizations, such as EY and MLA (beef supply chain projects), owners of the Australian Miles (biomass and bioenergy assessment—sugarcane projects), and Brisbane Royal Hospital (health system project). He has a wide range of experience in academic research and industrial projects with more than 80 refereed journals, conference papers, and industrial reports.