

IoT-Based Route Recommendation for an Intelligent Waste Management System

Mohammadhossein Ghahramani¹, *Member, IEEE*, MengChu Zhou², *Fellow, IEEE*,
Anna Molter, and Francesco Pilla¹

Abstract—The Internet of Things (IoT) is a paradigm characterized by a network of embedded sensors and services. These sensors are incorporated to collect various information, track physical conditions, e.g., waste bins' status, and exchange data with different centralized platforms. The need for such sensors is increasing; however, the proliferation of technologies comes with various challenges. For example, how can IoT and its associated data be used to enhance waste management? In smart cities, an efficient waste management system is crucial. Artificial intelligence (AI) and IoT-enabled approaches can empower cities to manage the waste collection. This work proposes an intelligent approach to route recommendation in an IoT-enabled waste management system given spatial constraints. It performs a thorough analysis based on AI-based methods and compare their corresponding results. Our solution is based on a multiple-level decision-making process in which bins' status and coordinates are taken into account to address the routing problem. Such AI-based models can help engineers design a sustainable infrastructure system.

Index Terms—Artificial intelligence (AI), evolutionary algorithms, optimization, route recommendation, smart cities, waste management.

I. INTRODUCTION

THE WORLD'S population living in urban areas is expected to continue to grow at a fast pace. Such population growth can lead to an increasing waste generation. Today, big cities worldwide face various waste management challenges due to rapid growth in population and consumption. Generally speaking, waste management consists of different procedures, such as garbage collection, transport, processing, waste disposal, and monitoring [1]. The challenges in this field include insufficient infrastructure for waste collection, lack of financial resources, health issues, and proper waste management planning [2]. In this work, we deal with waste collection and route optimization. A waste management process is explicitly identified as a service. It consumes a significant portion

of a municipality's operating budget. A considerable amount of this budget goes on garbage collection and transportation. The scenario seems to be an unreasonable wastage of resources if bins are collected while partially filled up. To deal with this situation, bins should be built on a microcontroller-based platform to be monitored effectively. In this way, the information about the status of bins (i.e., the level of garbage in bins) can be obtained before dispatching garbage collector vehicles for garbage collection. The conventional way of collecting garbage is a cumbersome procedure requiring much individual effort, time, and cost [3]. Developing new enabling technologies can help alleviate the situation. Artificial intelligence (AI) has made its way into different areas, e.g., engineering [4], [5], urban planning [6], and management [7]. AI-based models are incorporated in different fields of study, such as manufacturing, to solve various problems, such as process planning, route optimization, and extracting insights from sensor data [8]–[11]. AI and Internet of Things (IoT) technologies can pave the way to develop an efficient system and automate waste management practices. They offer new strategies for reducing the cost, and the complexity of waste management [12].

AI-based approaches, such as the genetic algorithm (GA) can be adopted to increase efficiency and reduce cost [13]. Their applications to waste management can make garbage collection processes smart. Such methods can substitute traditional ones for planning routes for waste collection while using fewer resources in transportation. Route planning in the context of waste management is defined as the process of identifying and evaluating the feasibility of routes for collecting garbage efficiently. This process includes different phases, e.g., identifying pickup locations, clustering locations/bins, measuring the amounts of waste to be collected, and planning a route [14]. Proposing an efficient and practical waste collection approach is consistent with the characteristics of a given problem. In this work, we develop our solution based on the inherent properties of the concerned problem using evolutionary algorithms. The problem is formulated and modeled based on two different scenarios, i.e., discrete optimization [15] and continuous one [16]. Spatial constraints are integrated into the implemented models. They are employed to address proximity issues to be explained later. First, we explain how a routing problem is modeled based on both approaches. Then, we compare them and show the former (i.e., the proposed discrete optimization model) can provide a much better solution than the latter. Finally, different discrete algorithms are

Manuscript received August 9, 2021; revised November 10, 2021; accepted November 29, 2021. Date of publication December 2, 2021; date of current version July 7, 2022. This work was supported in part by the Science Foundation Ireland under Grant 16/IA/4610. (Corresponding author: Mohammadhossein Ghahramani.)

Mohammadhossein Ghahramani, Anna Molter, and Francesco Pilla are with the Spatial Dynamics Lab, University College Dublin, Dublin 4, D04 V1W8 Ireland (e-mail: sepehr.ghahramani@ucd.ie; anna.molter@ucd.ie; francesco.pilla@ucd.ie).

MengChu Zhou is with the Helen and John C. Hartmann Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Digital Object Identifier 10.1109/JIOT.2021.3132126

compared with the proposed discrete model to validate the result.

All bins are built on a microcontroller-based platform, embedded with sensors informing about the level of garbage using an application programming interface (API). Our solution includes several phases. The first phase deals with defining different fullness levels, e.g., 60% capacity of a given bin. Whenever bins come to this predefined level, they are given a priority upon their collection. Moreover, those bins whose capacity does not reach a minimum level (e.g., 20% capacity) are not considered in the second phase of the algorithm. It should be mentioned that considering this flexibility can reduce the complexity of the algorithm significantly. The status of bins is obtained by querying the interface using API. The second phase concerns planning the best path given the bins' status determined in the prior phase. Dealing with spatial optimization problems is more challenging than nonspatial ones. Topological constraints should be considered when solving a spatial optimization problem; in other words, the physical properties of spatial points and their neighbors should be considered. The main contributions of an efficient spatial optimization method include computational efficiency, flexibility, and optimization quality. In line with the mentioned goals, we propose an approach that improves computational efficiency and achieves much higher optimization quality than existing ones. The contributions of this work are as follows.

- 1) An optimized IoT-based model including intelligent vehicle routing strategies coupled with spatial constraints is proposed to offer promising search capability in a discrete and continuous domain for waste management.
- 2) The model can maintain diverse solutions given different decision-making constraints for sustainability concerns.

The remainder of this article is organized as follows. Some related work about the models that have been designed for waste collection is described in Section II; both implemented models are discussed in Section III; the experimental settings and the comparison results are shown in Section IV; and the future work and conclusions are presented in Section V.

II. RELATED WORK

Recent studies have been focused on different ways to exploit the opportunities offered by AI and machine learning due to the limitations of conventional computational approaches [17]. In environmental engineering, leveraging IoT sensors has enabled access to data that can be traced throughout various planning phases, from higher level strategic planning processes to lower level operational planning. AI-based techniques, such as artificial neural network (ANN), multilayer perception, adaptive neuro-fuzzy inference system, and evolutionary algorithms have been implemented to address air pollution and emissions reduction [18]–[20]. Moreover, different technologies, such as cloud and edge computing [21]–[25], are also embedded in today's urban infrastructures [26]–[29]. These technologies are used to store and process data from almost anywhere. They are being adopted across various enterprises to improve operations and develop better analytics to

enhance decision-making ability. Given these technologies, intelligent data-driven mechanisms can be developed to enable authorities to manage waste effectively.

Broadly speaking, route optimization is referred to as an NP-Hard problem. Therefore, most studies in this context are based on heuristic and metaheuristic algorithms [30]–[34]. However, metaheuristics can provide more robust, effective, and cost-efficient solutions to multiobjective problems. The quality of solutions provided by these algorithms is much higher than those obtained by conventional heuristic ones [35]. Hence, metaheuristics have received considerable attention for solving problems. Wy and Kim [36] developed a heuristics method to model a rollon–rolloff waste collection vehicle routing problem. To do so, they propose a neighborhood search-based iterative heuristic approach. In [37], a parallel metaheuristic approach based on simulated annealing (SA) is proposed to incorporate asynchronous and synchronous Markov chains. A residual capacity and radical surcharge algorithm have been considered to generate an initial solution for the model; then, local search methods are implemented to optimize the obtained solution. These models can be computationally expensive since heuristic and metaheuristic algorithms are integrated. Wang and Lu [38] have introduced a memetic algorithm with competition to solve the routing problem. Their solution is based on a k -nearest neighbor approach coupled with an SA strategy. First, they have implemented a permutation-based method to encode the solution. Then, an effective decoding method has been constructed. Finally, SA has been used to find optimal routes. Different variants of the classical SA algorithm have also been adopted for route optimization in [39] and [40]. However, a sequence of collection tasks has not been considered in these studies. To alleviate this concern, an optimization problem is solved by using metaheuristics such that the number of collection trips is minimized [41]. The work [41] provides a theoretical model for a routing problem with the characteristics of full loads and multiple trips. A traveling salesman problem [42], as a combinatorial optimization problem, has been considered to model vehicle routing problems [43], [44]. For these problems, diverse optimal solutions should be provided. GA has been used in waste management planning to model waste accumulation, facility siting, and generation [45]. Amal *et al.* [46] have proposed a GIS-based GA for optimizing the route of solid waste collection. They use a modified version of the original Dijkstra algorithm in GIS to generate optimal solutions. They have also conducted a case study at Sfax city in Tunisia to validate the performance of their method.

Several approaches to route planning for waste management have been proposed in the literature. Some of them are based on periodic collection [36]; while in some others, the collection procedure is done within a specific time window [39]. Collection approaches can vary according to urban characteristics, i.e., residential or commercial/industrial areas. In residential areas, the collection is primarily on a door-to-door basis and carried out by vehicles passing along the streets to collect accumulated garbage from households. In commercial districts, trucks visit different locations, e.g., designated business locations, such as hotels and shopping malls or waste

containers in industrial regions, to collect garbage. Despite the advantages of the existing models, they are not efficient if used to solve our concerned problem. They cannot appropriately model the associations among spatial objects. Effective solutions are highly dependent on the characteristics of problems to be solved. For example, we deal with proximity issues in this work. To address such concerns, spatial constraints should be integrated into our approach. This work aims to present an efficient approach, given the problem characteristics while addressing all related concerns. It also compares its results with other methods.

III. WASTE COLLECTION MODELS

The proposed IoT-based model is based on integrated algorithms, including GA and an ANN. The associated cost of the model is calculated by incorporating ANN into the model. Generally speaking, GA consists of different phases, i.e., parent selection, crossover, mutation, and creating the final population [9]. Parent selection is a vital phase, consisting of a finite repetition of various procedures, such as selecting parent strings, recombining strings, and mutation operations. The reproductive phase's goal is to choose cost-efficient individuals from the population and produce new offsprings for the next generation. An effective mechanism should be considered to deal with the exploration and exploitation of an algorithm and avoid premature convergence. The mentioned concerns can lead to a loss of diversity. An efficient solution should also eliminate the cost scaling issue and adjusts the selection procedure. All related concerns have been addressed in this work. We consider a selection pressure approach and adjust the balance between exploration and exploitation by recombining crossover operators to adjust their probabilities. Consequently, individuals are produced throughout the mating pool by establishing a hybrid roulette tournament pick operator. A discussion for determining the exploration and exploitation rate is presented throughout this article. Given the proposed model, garbage collectors can be sent to respective locations to collect garbage based on all bins' status. The implemented approach is practicable for different route planning.

A. Spatial Constraints

The information about bins is obtained by querying API of a waste management company (i.e., Bigbelly bins and Futurestreet) based in Dublin, Ireland. The model interactively submits hypertext transfer protocol (HTTP) requests, and the server returns responses. The response contains spatial and nonspatial information, coordinates of bins in Docklands (i.e., a vibrant area in Dublin), and their corresponding fullness status (Fig. 1). The parameters of HTTP requests, such as API Token, Station IDs, StartTime, and EndTime are also contained. We deal with spatial characteristics and associated complexities in this work. Spatial analysis can be much more challenging than traditional data processing due to the complexity of possible patterns. In such an analysis, observations should be defined based on spatial relationships, e.g., spatial adjacency [47]. Different methods (e.g., graph-based, grid

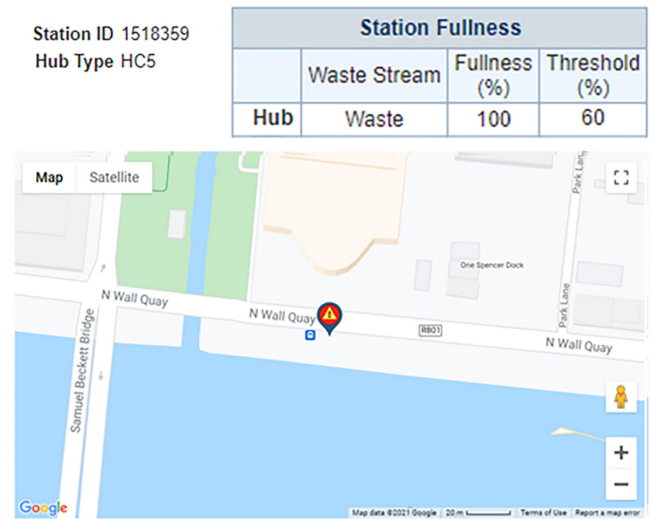


Fig. 1. Querying API to check the status of waste bins.

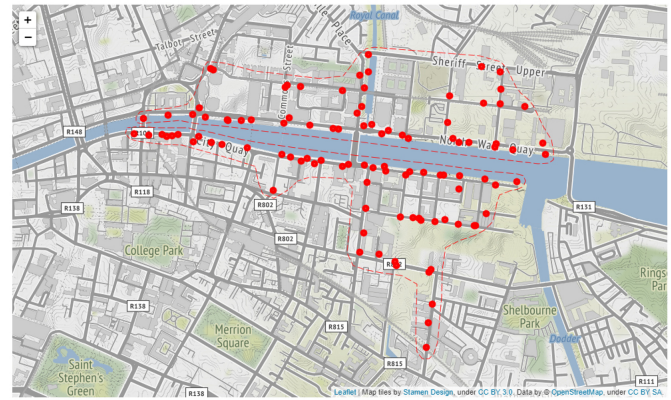


Fig. 2. Distribution of bins in Docklands, Dublin.

neighbors, K -nearest neighbors, distance-based, and higher order methods) define adjacency for a point or areal data. Some of these techniques are based on interpoint distances, such as k -nearest neighbors, while others are based on neighbors of areal units. As explained, an effective spatial analysis should be developed based on such proximity measures. However, defining adjacency criteria depends on the characteristics of a given data and relies on the spatial distribution of observations (e.g., bins together with all related features). The distribution of bins in Docklands (the area where our concerned bins are located) is demonstrated in Fig. 2. As can be seen, a river runs through the area where the bins are located. In this scenario, defining a neighborhood list is challenging as finding shared boundaries among different areal units does not account for cross-area relationships segregated by bridges. Those units that are connected by bridges cannot be considered as neighborhood links. Therefore, we have implemented the neighborhood list used in the proposed algorithm based on a graph-neighbor approach to overcome such shortcomings. To that end, two different graph-based techniques, i.e., Delaunay triangulation and Gabriel graph methods [48], are used. We have found that the former results in a more robust outcome. By defining this

proximity measure, bins are automatically divided into two groups. As illustrated in Fig. 2, two groups of bins can be separated automatically.

B. Spatial Objective Function

Suppose that at n spatial data points (bins) $b_i, i \in 1, 2, \dots, n$, we observe $y_i = y(b_i)$. $y(b_i)$ is the status of the i th bin. This vector can be defined as

$$Y = [y_1, y_2, \dots, y_n]^T. \quad (1)$$

Then, each b_i is defined as: $b_i = (\text{longitude}; \text{latitude}; Y_i; C_i)$, including the coordinates of all bins in Docklands, Y_i is the associated percentage of fullness of the i th waste bin b_i , and C_i represents the cluster that the i th bin belongs to. It is worth mentioning that each b_i is implemented as a *spatial data point*, consisting of different slots, such as projection-string, plot-order, and bounding-box. These features are used for projection purposes. The defined spatial data points are the objects of coordinate reference systems.

The spatial-weights-matrix W is also defined by considering all nearby bins as

$$W = [w_{ij}]_{n \times n} \quad (2)$$

the weights between pairwise adjacent bins b_i and b_j satisfy

$$w_{ij} > 0 \text{ for any adjacent bins, and } w_{ii} = 0 \text{ for any } i. \quad (3)$$

Let $G = (V, E)$ be an *undirected weighted graph* with n bins, where $V = \{V_1, V_2, \dots, V_n\}$ is a set of spatial nodes, and E is a matrix representing connecting edges (indicating the path between two nodes), i.e.,

$$E = [E_{ij}]_{n \times n} \quad (4)$$

where i and j represent bins i and j . The distance among bins is also considered as the edge's length (e.g., the weight value of edge (i, j) is denoted as d_{ij}). The desired path (the algorithm's objective function) can be formulated as the permutation Θ

$$\min f(\Theta) = \sum_{k=1}^{n-1} d_{\Theta(k)\Theta(k+1)} + d_{\Theta(n)\Theta(1)} \quad (5)$$

where $\Theta(k)$ is the k th element of permutation Θ .

Based on the defined list of bins and their pairwise distances, the aim is to search for the shortest path in a weighted graph such that each bin is visited exactly once. This problem is an NP-hard optimization problem and computationally expensive since the number of permutations of n nodes grows exponentially with n . Given the nature of this problem, it is challenging to propose a solution based on deterministic algorithms. Hence, we propose two different approaches based on metaheuristic techniques to obtain near-optimal solutions. The implemented techniques are population-based models coupled with metaheuristic search algorithms. Such approaches enable us to find a near-optimal solution within a reasonable time while avoiding the need for exhaustively exploring the input space. It should be mentioned that most studies have been focused on proposing solutions based on a continuous optimization domain. We aim to find an appropriate algorithm so that it can offer promising search capability and

maintain diverse solutions. Since we deal with a combinatorial optimization problem, a GA is selected. It should be noted that some of the evolutionary algorithms, such as particle swarm optimization (PSO) or covariance matrix adaptation evolutionary strategies, are originally proposed for a continuous search space [49].

GA includes a repetition of different operations sequences, i.e., parent selection, recombination, and mutation. We implement two variants of GA (with different genetic operators) in continuous and discrete cases. We show that the one conducting for the discrete space performs much better. The models treat all bins (chromosomes in GA terminology) as bit-strings. Both models rely on a population of individuals (candidate solutions) to explore a search space. These candidates are a set of chromosomes and encoded as strings. The models use an initial population and genetic operators, such as crossover and mutation. These operations are implemented to produce a new generation by recombining a population's chromosomes. Then, fitter individuals are chosen, given an objective function. The operators are also utilized for exploring and exploiting search space. It is of great importance to balance the exploration and exploitation of an algorithm. Exploring the input space is done to find an optimal solution by using a crossover operation. Besides, a mutation operator is considered to avoid the premature convergence of the algorithm. We have controlled the exploration/exploitation level by a selection pressure parameter (β in this work). This parameter β is used in the parent selection phase. All the procedures, iteratively, are repeated until some termination criteria are met. The best solution (the one with the minimum cost when the algorithm ends) is then selected.

As explained, the goal is to find the best route for waste collection. The first step is to define an initial population. Then, dynamic crossover and mutation operators are developed. The defined spatial objective function is used to minimize the total traveled distance. The procedures are described for both scenarios next.

C. Implemented Model: Discrete Scenario

In this scenario, a discrete GA coupled with the defined spatial objective function is implemented. To that end, an initial population is randomly generated (given the algorithm's population size). The cost associated with each chromosome is measured based on the defined objective function. These costs should be minimized. The initial population is defined as individuals, including n -dimensional chromosomes. A path set (ζ) can be defined as a permutation of n bins

$$\zeta = (b_1, b_2, \dots, b_n). \quad (6)$$

Algorithm 1 represents the way in which the initial population is created.

Then, cost-efficient chromosomes from the population are sorted and selected. A parent selection phase is employed for creating a new population at each iteration. In the reproduction phase, permutative crossover and permutative mutation operators are implemented. These operations are to be explained later. All the procedures are repeated until a termination criterion is met. A feasible solution to the problem satisfies the

Algorithm 1: Pseudocode for Creating Initial Population Given Their Cost (Discrete Scenario)

Input : Spatial-Data-Frame (b_1, b_2, \dots, b_n)
 $Dist \leftarrow$ distance function

Output: $cost \leftarrow$ fitness of each permutation

```

1  $n \leftarrow$  number of bins
2  $P \leftarrow []$ 
3  $\eta_p \leftarrow$  Size of population
4  $L \leftarrow 0$ 
5 Select bins given their level of fullness
6 # CostFn
7 for  $i \leftarrow 1, \dots, \eta_p$  do
8    $P_i.position \leftarrow$  a random permutation
9    $P_i.cost \leftarrow$  calculate cost given:
10  for  $i \leftarrow 1, \dots, n-1$  do
11     $cost.append \leftarrow$  cost of each permutation:
12     $L = L + Dist(P_i, P_{i+1})$ 
13  end
14 end
15  $P \leftarrow$  Sort population ( $P$ );
16 return initial individuals;
```

discussed constraints. Let F be the set of all solutions; the aim is to find an optimal one ($f^* \in F$) given the cost function. In another word, we are looking for a permutation Θ^* of n bins (b_1, b_2, \dots, b_n) such that

$$\sum_{k=1}^n d(b_{\Theta^*(k)}, b_{\Theta^*(k+1)}) \leq \sum_{k=1}^n d(b_{\Theta(k)}, b_{\Theta(k+1)}). \quad (7)$$

The cost of each solution is calculated by using an ANN model. These costs are used for different purposes like parent selection and then the solution. The objective in the parent selection phase is to select two solutions (with lower costs) from the population such that newly created offsprings would inherit their parents' characteristics. There are different methods to select parents, e.g., random selection, rank selection, stochastic universal sampling (SUS), tournament selection, and Boltzmann selection. We do not adopt the random selection since it has no selection pressure parameter. Rank selection and SUS suffer from premature convergence. Applying these methods can easily lead to a local optimum. Instead, we have employed Boltzmann selection. This method is inspired by SA and can help maintain a good diversity. The probability of a solution being chosen is measured, given the following Boltzmann probability:

$$P(i) = \frac{e^{-\beta L_i}}{\sum_{k=1}^{\eta_p} e^{-\beta L_k}} \quad (8)$$

where η_p is the size of the initial population, and L is the defined cost function. β is the selection pressure. Parents are determined according to probabilities, which are proportional to the costs calculated earlier. In other words, solutions with a lower cost are more likely to be selected than those with a greater one. It is worth mentioning that β is selected, such that $\sum_{i \in H} P(i) = 0.7$, where H is the set of half of the best solutions (population is sorted according to their cost values, and $\eta_p/2$ of them are selected). Consequently, the Roulette Wheel mechanism is implemented for sampling. This process is repeated until a predefined number of parents are selected.

Algorithm 2: Pseudocode for Finding an Optimal Path (Discrete Scenario)

Input : CostFn,
 $P \leftarrow$ Initial Population

Output: Optimal Path

```

1  $I^{Max} \leftarrow$  Maximum Iterations
2  $\theta \leftarrow$  Crossover rate
3  $\eta_c \leftarrow$  Size of crossover (based on  $\theta$ )
4  $\eta_m \leftarrow$  Size of mutation
5 for  $i \leftarrow 1, \dots, I^{Max}$  do
6   # Crossover operation
7    $Pr(s \in P) = \frac{\exp^{(-\beta) * \frac{L_s}{LargestCost}}}{\sum_{k=1}^{\eta_p} \exp^{(-\beta) * \frac{L_k}{LargestCost}}}$ 
8   for  $i \leftarrow 1, \dots, \eta_c$  do
9     Select two parents ( $\gamma$ ) based on the RW function;
10    Generate two offsprings ( $\alpha$ );
11    [ $\alpha(i, 1).position, \alpha(i, 2).position$ ]  $\leftarrow$ 
      CrossoverFn( $\gamma 1.position, \gamma 2.position$ )
12    [ $\alpha(i, 1).cost, \alpha(i, 2).cost$ ]  $\leftarrow$ 
      CostFn( $\gamma 1.position, \gamma 2.position$ )
13  end
14
15  # Mutation operation
16  for  $i \leftarrow 1, \dots, \eta_m$  do
17    Select one parent ( $\gamma$ ) based on the RW function;
18    [ $\alpha(i).position$ ]  $\leftarrow$  MutationFn( $\gamma.position$ )
19    [ $\alpha(i).cost$ ]  $\leftarrow$  CostFn( $\gamma.position$ )
20  end
21   $P \leftarrow$  [P, Offsprings, and Mutants];
22   $P \leftarrow$  Sort and select first  $\eta_p$  individuals;
23   $S^* \leftarrow P(1).position$ ;
24   $Cost^*(i) \leftarrow P(1).cost$ ;
25 end
26 return optimal path;
```

Therefore, solutions with the greatest cost have a minimal chance of being chosen. These procedures are described in Algorithm 2.

After selecting parents, a crossover operation should be employed. On this basis, the chromosomes of selected parents are combined to create new offspring. We deal with a combinatorial problem; hence, traditional methods (such as one-point, two-point, and uniform crossovers) for conducting the crossover operation are not suitable. Two cut points on each pairwise parent have been considered in order to create a permutative crossover. The portion after each cut point has been selected and exchanged (the selected bit strings of the first parent are mapped onto the other parent's string). All bits are then checked to ensure there are no conflict bits to guarantee each chromosome is permutation without repetition. In this way, new offsprings are created. To avoid being trapped in a local minimum, a mutation operator has been considered. It also maintains a good diversity in a newly generated population. Different types of mutation operations (e.g., insertion, inversion, and swap) have been considered in our model. Generally, one or more bits of chromosomes are replaced, such that generated offsprings also maintain permutation patterns. The pseudocode realizing the algorithm operators is given in Algorithm 3.

As stated, the goal is to minimize cost function L . To that end, an ANN has been integrated into the model. The model

Algorithm 3: Pseudocode for Defining the Algorithm's Operators (Discrete Scenario)

Input : Two parents, (γ)
Output: Offsprings

```

1 Function [O1, O2] = CrossoverFn( $\gamma_1, \gamma_2$ )
2   | CrossoverMethod  $\leftarrow$  {One-Point Crossover}
3   | return two offsprings;
4 end
5 Function M = MutationFn( $\gamma$ )
6   | Apply three mutation operators (Swap, Inversion, and
7   | Insertion);
8   | Prswap  $\leftarrow$  probability of using Swap operator;
9   | Prinsertion  $\leftarrow$  probability of using Insertion operator;
10  | Prinversion  $\leftarrow$  probability of using Inversion operator;
11  | Method  $\leftarrow$  RW(Prswap, Prinsertion, Prinversion);
12  | switch Method do
13  |   | case 1 do
14  |   |   | Swap( $\gamma$ )
15  |   | end
16  |   | case 2 do
17  |   |   | Insertion( $\gamma$ )
18  |   | end
19  |   | case 3 do
20  |   |   | Inversion( $\gamma$ )
21  |   | end
22  | end
23  | return mutant;
24 end

```

gets the objective function (i.e., L) as an input. Then, the ANN is used to measure corresponding cost values in different iterations. Iteratively, different solutions (permutations) are generated and evaluated by GA's operations. A Levenberg–Marquardt training algorithm consisting of two layers (15 neurons in the hidden layer) is utilized. It enjoys adaptive weights with full connectivity among neurons in the input and hidden layers. All costs are calculated, and the path is selected such that the corresponding cost is minimized.

D. Implemented Model: Continuous Scenario

In this scenario, we implement GA that leverages random keys to encode solutions. Despite the method used above (where a stream of integers is used to represent the order in which bins are to be visited), in this model, each bit string in a chromosome is assigned with a random number drawn uniformly from $[0, 1)$ and sorted in the ascending order. Hence, each solution has two different parts, i.e., an integer part and a fractional one. The former refers to what we used for permutation, while the latter indicates random key numbers assigned to bit strings. The initial population is generated by creating n chromosomes, where n is the number of bins. Then, a real number drawn randomly from $[0, 1)$ is given to each bin. New individuals are spawned in the algorithm's reproduction phase given a predefined crossover rate and mutation one. The procedure realizing the initial population generation is presented in Algorithm 4.

After the initial population is generated, different operators are performed for passing the best solutions in the population to the next generation based on an elitist strategy. First, two parents are selected, and a parameterized single crossover is

Algorithm 4: Pseudocode for Creating Initial Population Given Their Cost (Continuous Scenario)

Input : Spatial-Data-Frame (b_1, b_2, \dots, b_n)
 $Dist \leftarrow$ distance function
Output: cost \leftarrow fitness of each solution

```

1 Select bins given their level of fullness
2  $n \leftarrow$  number of bins
3  $\eta_p \leftarrow$  Size of population
4  $S \leftarrow n$  uniformly random numbers in the interval (0,1)
5 # Turn a real vector into a permutation
6  $S \leftarrow \text{sorted}(S)$ 
7  $L \leftarrow 0$ 
8 # CostFn
9 for  $i \leftarrow 1, \dots, n-1$  do
10  | cost.append  $\leftarrow$  cost of each permutation:
11  |  $L = L + Dist(S_i, S_{i+1})$ 
12 end
13 return solutions and costs;

```

Algorithm 5: Pseudocode for Finding Optimal Path (Continuous Scenario)

Input : CostFn,
 $P \leftarrow$ Initial Population
Output: Optimal Path

```

1  $i \leftarrow 0$ 
2 Randomly initial population ( $P_i$ )
3 Evaluate each solution in  $P_i$ 
4  $\eta_p \leftarrow$  Size of population
5 Repeat
6 for  $j \leftarrow 1, \dots, \eta_p$  do
7   | Select two parents from ( $P_i$ )
8   |  $P_j = \text{CrossoverFn}(P_1, P_2)$ 
9   |  $P_j = \text{MutationFn}(P_j)$ 
10  |  $P_{i+1} \leftarrow P_j$ 
11 end
12  $i \leftarrow i + 1$ 
13 return optimal path;

```

considered for producing two new offsprings. In the defined crossover, one random position in the array of bits is selected and exchanged. As discussed earlier, after the initial population is generated, the parent selection operation should be conducted in the reproduction phase. The objective is to choose solutions with minimal cost. The cost function and the ANN used for calculating costs are both the same as those described in the previous section. The newly added offsprings inherit bits based on a probability measure discussed earlier. Second, a mutation operator is implemented to ensure a diverse population. It should be mentioned that each time a new solution is generated, it is compared to previous solutions in the population to make sure no duplicate individuals are maintained. The maximum number of iterations (the algorithm termination criterion) is also set to 1000. Algorithm 5 realizes the elitist strategy in the random-key approach.

IV. RESULTS

In this work, we have proposed an optimized IoT-based waste collection model, including intelligent vehicle routing strategies coupled with spatial constraints. The goal is to present an efficient and practical waste collection approach

TABLE I
RESULTS OF THE DISCRETE OPTIMIZATION MODEL, GIVEN DIFFERENT PARAMETER SETTINGS

Crossover Rate	Population Size	Neurons	Corresponding cost for selected solution in different iterations											
0.6	70	6	1.2031	1.2009	1.1702	1.1699	1.1664	...	0.3212	0.3212	0.3178	0.3178	0.3178	0.3178
		10	1.2102	1.1002	1.1002	1.1002	1.0847	...	0.2915	0.2905	0.2833	0.2830	0.2765	0.2765
		15	1.3202	1.2723	1.2721	1.2621	1.2586	...	0.3082	0.3080	0.3003	0.3003	0.3003	0.3003
0.6	80	6	1.3124	1.3012	1.2731	1.1974	1.1844	...	0.2523	0.2475	0.2378	0.2378	0.2370	0.2370
		10	1.3012	1.1302	1.1922	1.1804	1.1543	...	0.2814	0.2805	0.2805	0.2805	0.2711	0.2711
		15	1.2952	1.2653	1.2653	1.2653	1.2641	...	0.3051	0.3050	0.2976	0.2976	0.2936	0.2936
0.6	90	6	1.4142	1.3839	1.2943	1.2602	1.2461	...	0.2903	0.2785	0.2785	0.2785	0.2785	0.2785
		10	1.3772	1.1372	1.1362	1.1361	1.1347	...	0.2664	0.2508	0.2508	0.2449	0.2443	0.2443
		15	1.3912	1.3644	1.2983	1.2703	1.2699	...	0.3001	0.2751	0.2707	0.2707	0.2700	0.2700
0.7	70	6	1.0921	1.0622	1.0613	1.0613	1.0509	...	0.2071	0.2054	0.2054	0.1963	0.1963	0.1963
		10	1.0903	1.0741	1.0608	1.0573	1.0491	...	0.2121	0.2004	0.1864	0.1664	0.1663	0.1663
		15	1.0873	1.0873	1.0871	1.0776	1.0601	...	0.2215	0.2215	0.2035	0.1914	0.1843	0.1843
0.7	80	6	1.0801	1.0801	1.0773	1.0654	1.0491	...	0.2005	0.2001	0.1803	0.1704	0.1703	0.1703
		10	1.0691	1.0685	1.0611	1.0464	1.0459	...	0.1822	0.1820	0.1820	0.1614	0.1612	0.1612
		15	1.0673	1.0572	1.0417	1.0417	1.0417	...	0.1726	0.1710	0.1703	0.1702	0.1702	0.1702
0.7	90	6	1.0734	1.0732	1.0549	1.0567	1.0501	...	0.1819	0.1810	0.1753	0.1715	0.1711	0.1711
		10	1.0601	1.0579	1.0449	1.0447	1.0441	...	0.1603	0.1602	0.1602	0.1602	0.1602	0.1602
		15	1.0626	1.0567	1.0549	1.0402	1.0402	...	0.1653	0.1649	0.1644	0.1621	0.1621	0.1621
0.8	70	6	0.8832	0.8803	0.8773	0.8672	0.8563	...	0.1321	0.1304	0.1304	0.1301	0.1301	0.1301
		10	0.8737	0.8705	0.8705	0.8644	0.8578	...	0.1316	0.1294	0.1293	0.1281	0.1280	0.1280
		15	0.8839	0.8812	0.8752	0.8638	0.8557	...	0.1325	0.1311	0.1311	0.1311	0.1311	0.1311
0.8	80	6	0.8124	0.8049	0.7991	0.7985	0.7961	...	0.1104	0.1101	0.1095	0.1095	0.1094	0.1094
		10	0.7856	0.7819	0.7751	0.7685	0.7661	...	0.1024	0.1024	0.1022	0.1021	0.1021	0.1021
		15	0.8094	0.8091	0.7977	0.7881	0.7763	...	0.1088	0.1081	0.1075	0.1075	0.1068	0.1068
0.8	90	6	0.8211	0.8026	0.8026	0.7988	0.7981	...	0.1164	0.1143	0.1121	0.1121	0.1120	0.1120
		10	0.8193	0.8109	0.7966	0.7965	0.7953	...	0.1105	0.1105	0.1086	0.1086	0.1086	0.1086
		15	0.8214	0.8052	0.7888	0.7885	0.7883	...	0.1109	0.1107	0.1095	0.1095	0.1091	0.1091
0.9	70	6	1.0874	1.0838	1.0775	1.0663	1.0661	...	0.1578	0.1575	0.1572	0.1572	0.1571	0.1571
		10	1.0893	1.0882	1.0725	1.0653	1.0565	...	0.1562	0.1561	0.1561	0.1559	0.1559	0.1559
		15	1.0843	1.0832	1.0819	1.0817	1.0811	...	0.1616	0.1610	0.1608	0.1608	0.1608	0.1608
0.9	80	6	1.0876	1.0775	1.0775	1.0696	1.0672	...	0.1408	0.1407	0.1407	0.1388	0.1388	0.1388
		10	1.0802	1.0801	1.0795	1.0763	1.0697	...	0.1475	0.1475	0.1472	0.1472	0.1471	0.1471
		15	1.0818	1.0808	1.0784	1.0678	1.0645	...	0.1463	0.1460	0.1460	0.1457	0.1457	0.1457
0.9	90	6	1.0823	1.0821	1.0805	1.0766	1.0751	...	0.1474	0.1462	0.1462	0.1458	0.1458	0.1458
		10	1.0831	1.0812	1.0705	1.0693	1.0651	...	0.1398	0.1398	0.1397	0.1395	0.1394	0.1394
		15	1.0794	1.0764	1.0744	1.0701	1.0688	...	0.1418	0.1416	0.1415	0.1412	0.1411	0.1411

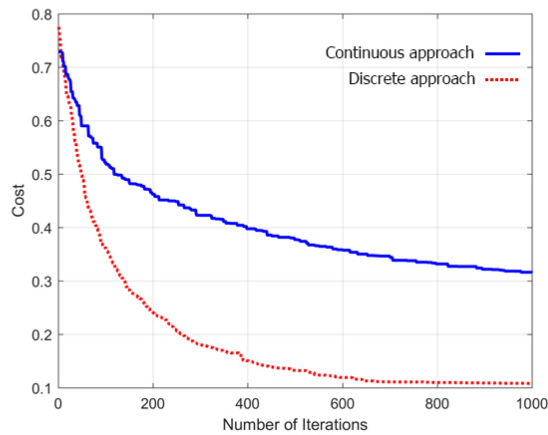


Fig. 3. Comparing the costs of both models in different iterations.

given the characteristics of the concerned bins data. Dublin City Council has partnered with Big Belly Bins to install 110 smart bins across the Dublin Docklands area. All bins are built on a microcontroller-based platform, embedded with sensors informing about the level of garbage. These sensors provide real-time information showing the capacity of each of the bins. This allows us to see how much waste each bin contains in terms of percentages, i.e., 60% or 80%. The proposed IoT-based solution defines the level of fullness, and the optimal path is determined. We have proposed two different scenarios based on a hybrid approach. All experiments are conducted on an 18-core system with 192-GB memory. Two implemented hybrid metaheuristic methods (integrated with ANN) evaluate various solutions to optimize the defined

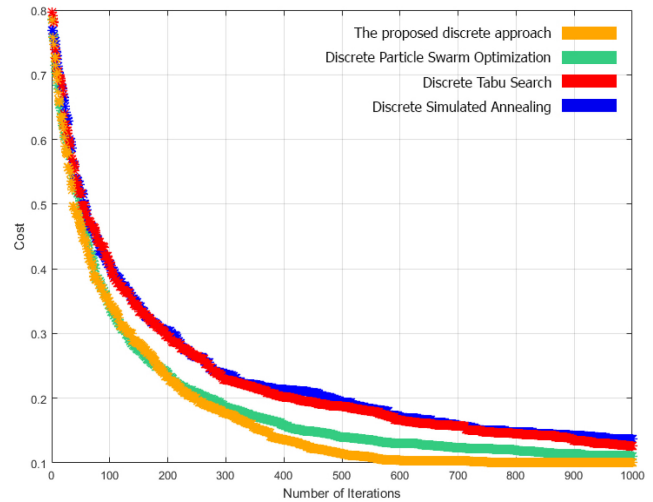


Fig. 4. Comparing different discrete algorithms.

objective function. The volume of the data set and the number of bins are considered for defining the initial population rate. The number of neurons is designated based on a trial-and-error method. Moreover, the volume of the concerned data set has been considered in defining the initial population rate. As stated, the cost of each solution is calculated by using an ANN model. These costs have been used for different purposes like parent selection. To that end, a multilayer perceptron with the Levenberg–Marquardt training algorithm has been used (since it converges faster and more accurately toward our problem) consisting of two layers of adaptive weights with

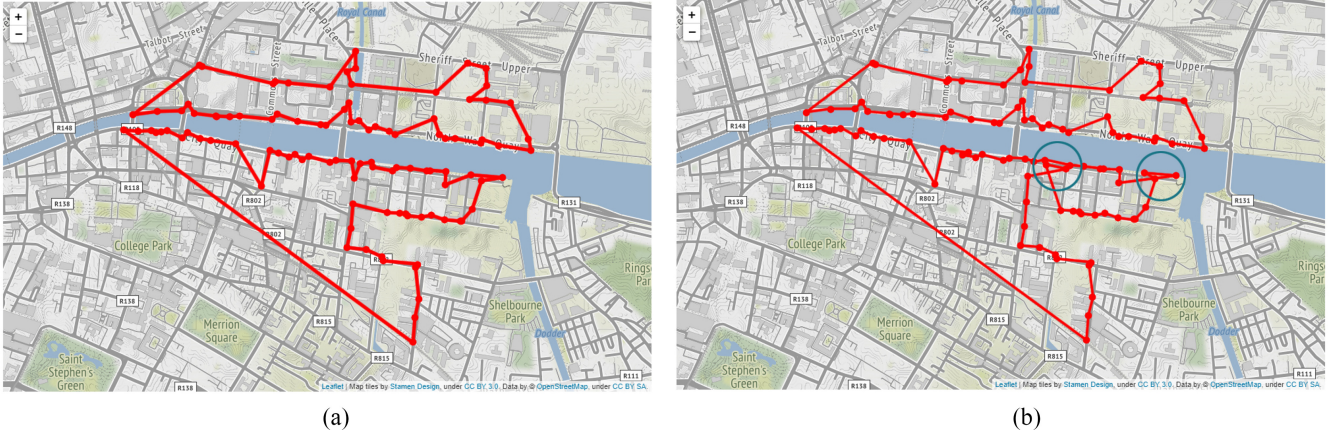


Fig. 5. Results of two implemented models after 1000 iterations, i.e., (a) optimal path, given the discrete optimization model and (b) optimal path, given the continuous optimization model.

TABLE II
RESULTS OF THE CONTINUOUS OPTIMIZATION MODEL, GIVEN DIFFERENT PARAMETER SETTINGS

Crossover Rate	Mutation Rate	Population Size	Neurons	Corresponding cost for selected solution in different iterations								
0.6	0.2	80	10	1.1171	1.1164	1.1153	1.1144	...	0.5383	0.5383	0.5381	0.5381
		100	15	1.1124	1.1102	1.1014	1.1003	...	0.5221	0.5217	0.5211	0.5211
		120	20	1.2134	1.2117	1.1137	1.1034	...	0.5761	0.5703	0.5703	0.5703
0.6	0.3	80	10	1.1106	1.1104	1.1093	1.1054	...	0.4987	0.4871	0.4854	0.4853
		100	15	1.1102	1.1088	1.1075	1.1028	...	0.4761	0.4754	0.4701	0.4680
		120	20	1.1113	1.1096	1.1095	1.1053	...	0.4786	0.4784	0.4784	0.4783
0.6	0.4	80	10	1.1152	1.1098	1.1082	1.1066	...	0.4996	0.4982	0.4982	0.4981
		100	15	1.1151	1.1062	1.1043	1.1003	...	0.4683	0.4681	0.4677	0.4674
		120	20	1.1161	1.1102	1.1084	1.1068	...	0.4799	0.4799	0.4799	0.4799
0.7	0.2	80	10	0.8501	0.8471	0.8432	0.8301	...	0.3919	0.3886	0.3844	0.3842
		100	15	0.8314	0.8311	0.8011	0.7804	...	0.3654	0.3634	0.3634	0.3633
		120	20	0.8332	0.8252	0.7962	0.7492	...	0.3493	0.3486	0.3484	0.3483
0.7	0.3	80	10	0.7511	0.7510	0.7432	0.7402	...	0.3413	0.3411	0.3411	0.3411
		100	15	0.7301	0.7255	0.7032	0.7024	...	0.3215	0.3214	0.3210	0.3210
		120	20	0.7577	0.7431	0.7355	0.7207	...	0.3449	0.3446	0.3443	0.3443
0.7	0.4	80	10	0.7985	0.7742	0.7651	0.7554	...	0.3816	0.3796	0.3776	0.3776
		100	15	0.7921	0.7523	0.7501	0.748	...	0.3843	0.3842	0.3842	0.3841
		120	20	0.7935	0.7746	0.7653	0.7512	...	0.3821	0.3812	0.3812	0.3810
0.8	0.2	80	10	0.8321	0.8244	0.8201	0.8107	...	0.3919	0.3919	0.3919	0.3918
		100	15	0.8113	0.8018	0.8003	0.7982	...	0.3893	0.3852	0.3842	0.3838
		120	20	0.8214	0.8112	0.8014	0.8004	...	0.3896	0.3896	0.3895	0.3894
0.8	0.3	80	10	0.8127	0.8105	0.8077	0.8017	...	0.3931	0.3911	0.3911	0.3910
		100	15	0.8121	0.8028	0.7984	0.7975	...	0.4013	0.3991	0.3987	0.3982
		120	20	0.8201	0.8201	0.8103	0.7953	...	0.3924	0.3924	0.3924	0.3924
0.8	0.4	80	10	0.8654	0.8444	0.8259	0.8233	...	0.3991	0.3991	0.3990	0.3989
		100	15	0.8626	0.8611	0.8527	0.8422	...	0.3952	0.3948	0.3947	0.3943
		120	20	0.8427	0.8412	0.8410	0.8410	...	0.3892	0.3887	0.3885	0.3883

full connectivity among neurons in the input and hidden layers. Note that ANN is only used for cost calculation; hence, we have not included much information about this phase. The model queries API, and the information of bins (including their locations and level of fullness) is obtained. The optimal route is then detected after a series of iterative computations (given the termination criterion, e.g., the number of iterations or computation time). Fig. 3 displays the cost values in each iteration for both models. As can be seen, the algorithm based on discrete optimization converges much faster than the continuous one. The experimental results and the impacts of different parameter settings for the discrete and continuous methods are presented in Tables I and II, respectively. We have also implemented several discrete evolutionary algorithms and compared their results with the proposed discrete approach. Fig. 4 reveals the results obtained from different discrete evolutionary algorithms and shows the effectiveness of our approach. The suggested paths resulting from both algorithms are demonstrated in Fig. 5. As can be seen in

this figure, the second approach [Fig. 5(b)] is trapped in an optimal local solution. As discussed throughout this article, the level of fullness of bins has also been taken into account. When collectors are dispatched to collect waste, the level of fullness of bins is monitored. Fig. 6 illustrates the result given such consideration at a certain time. The level of fullness of bins depicted in blue in Fig. 6(a) is less than 20%, and 30% for those in Fig. 6(b). Given all results, we can conclude that the proposed discrete model is superior to the continuous one.

V. CONCLUSION AND FUTURE WORK

We have proposed an IoT-based model coupled with spatial constraints to solve a route recommendation problem in this work. We have modeled and solved the concerned problem through two different scenarios, i.e., discrete and continuous optimization approaches. Both methods were validated on a case study, and the results were compared. Although most

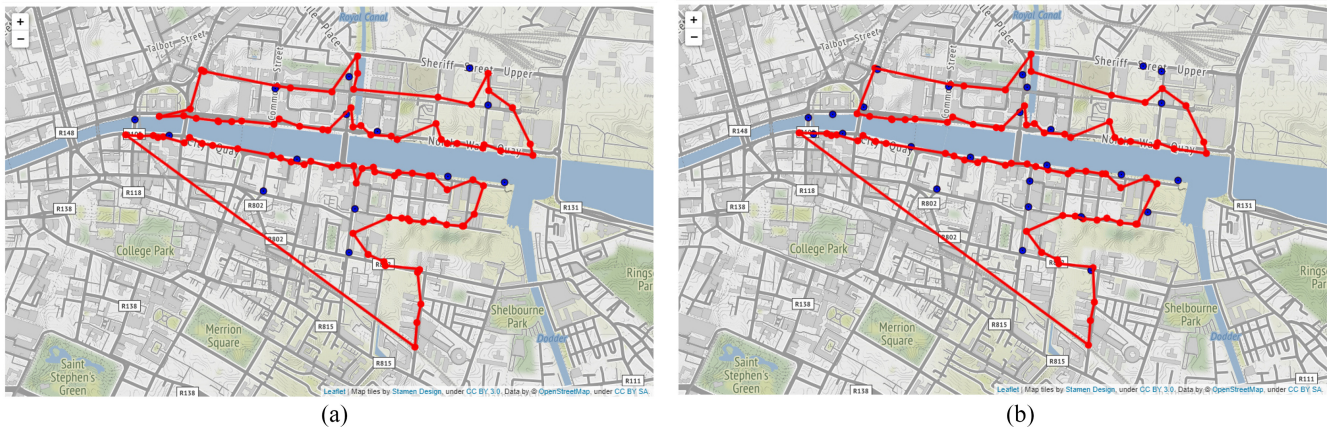


Fig. 6. Results of the discrete model given two different settings after 1000 iterations: (a) level of bins' fullness is 20% and (b) level of bins' fullness is 30%.

studies have focused on continuous optimization methods, we have shown that a discrete algorithm is more efficient in dealing with the problem. The latter can be easily trapped into the local optima. Different types of operators (e.g., permutation-based crossover and mutation operations) were designed to maintain diverse candidates during a search operation. We have also implemented some specific measures to control the balance between the exploration and exploitation of both algorithms. An ANN was utilized to calculate the associated cost in each iteration of the algorithms. Different discrete methods were also compared with our proposed discrete approach to validate the efficiency and effectiveness of the proposed model, and the results were illustrated. The main drawback of the state of the art was that it cannot appropriately model the associations among spatial objects, consequently find an optimal route. The proposed model can enable us to find an optimal solution within a reasonable time. It should be mentioned that the model can recommend multiple routes. A driver can then select the most favorable route from the suggested ones. In this work, we only obtained information about bins that are located in Docklands. As a part of our future work, we plan to consider all bins in Dublin.

ACKNOWLEDGMENT

The authors would like to acknowledge both Bigbelly and Futurestreet for providing the data used in this work.

REFERENCES

- [1] S. Ahmad, Imran, N. Iqbal, F. Jamil, and D. Kim, "Optimal policy-making for municipal waste management based on predictive model optimization," *IEEE Access*, vol. 8, pp. 218458–218469, 2020, doi: [10.1109/ACCESS.2020.3042598](https://doi.org/10.1109/ACCESS.2020.3042598).
- [2] Imran, S. Ahmad, and D. H. Kim, "Quantum GIS based descriptive and predictive data analysis for effective planning of waste management," *IEEE Access*, vol. 8, pp. 46193–46205, 2020.
- [3] Z. Wang, Y. Teng, H. Jin, and Z. Chen, "Urban waste disposal capacity and its energy supply performance optimal model based on multi-energy system coordinated operation," *IEEE Access*, vol. 9, pp. 32229–32238, 2021, doi: [10.1109/ACCESS.2021.3060421](https://doi.org/10.1109/ACCESS.2021.3060421).
- [4] W. Xiong *et al.*, "Wafer reflectance prediction for complex etching process based on K-means clustering and neural network," *IEEE Trans. Semicond. Manuf.*, vol. 34, no. 2, pp. 207–216, May 2021.
- [5] M. Ghahramani, M. C. Zhou, Y. Qiao, and N. Q. Wu, "Spatio-temporal analysis of mobile phone network based on self-organizing feature map," *IEEE Internet Things J.*, early access, Nov. 13, 2021, doi: [10.1109/JIOT.2021.3127203](https://doi.org/10.1109/JIOT.2021.3127203).
- [6] M. Ghahramani, N. J. Galle, C. Ratti, and F. Pilla, "Tales of a city: Sentiment analysis of urban green space in Dublin," *Cities*, vol. 119, Dec. 2021, Art. no. 103395.
- [7] M. Ghahramani and F. Pilla, "Analysis of carbon dioxide emissions from road transport using taxi trips," *IEEE Access*, vol. 9, pp. 98573–98580, 2021, doi: [10.1109/ACCESS.2021.3096279](https://doi.org/10.1109/ACCESS.2021.3096279).
- [8] L. Catarinucci, R. Colella, S. I. Consalvo, L. Patrono, C. Rollo, and I. Sergi, "IoT-aware waste management system based on cloud services and ultra-low-power RFID sensor-tags," *IEEE Sensors J.*, vol. 20, no. 24, pp. 14873–14881, Dec. 15, 2020, doi: [10.1109/JSEN.2020.3010675](https://doi.org/10.1109/JSEN.2020.3010675).
- [9] M. Ghahramani, Y. Qiao, M. C. Zhou, A. O'Hagan, and J. Sweeney, "AI-based modeling and data-driven evaluation for smart manufacturing processes," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 4, pp. 1026–1037, Jul. 2020.
- [10] M. Ghahramani, M. Zhou, and G. Wang, "Urban sensing based on mobile phone data: approaches, applications, and challenges," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 3, pp. 627–637, May 2020, doi: [10.1109/JAS.2020.1003120](https://doi.org/10.1109/JAS.2020.1003120).
- [11] M. Ghahramani, M. Zhou, and C. T. Hon, "Extracting significant mobile phone interaction patterns based on community structures," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1031–1041, Mar. 2019.
- [12] T. J. Sheng *et al.*, "An Internet of Things based smart waste management system using LoRa and Tensorflow deep learning model," *IEEE Access*, vol. 8, pp. 148793–148811, 2020, doi: [10.1109/ACCESS.2020.3016255](https://doi.org/10.1109/ACCESS.2020.3016255).
- [13] M. Ghahramani, A. O'Hagan, M. Zhou, and J. Sweeney, "Intelligent geodemographic clustering based on neural network and particle swarm optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, May 3, 2021, doi: [10.1109/TSMC.2021.3072357](https://doi.org/10.1109/TSMC.2021.3072357).
- [14] S. Ahmad, Imran, F. Jamil, N. Iqbal, and D. Kim, "Optimal route recommendation for waste carrier vehicles for efficient waste collection: A step forward towards sustainable cities," *IEEE Access*, vol. 8, pp. 77875–77887, 2020.
- [15] Y. Hou, N. Wu, M. Zhou, and Z. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 517–530, Mar. 2017.
- [16] A. Aldhfeeri and Y. Rahmat-Samii, "Brain storm optimization for electromagnetic applications: Continuous and discrete," *IEEE Trans. Antennas Propag.*, vol. 67, no. 4, pp. 2710–2722, Apr. 2019.
- [17] N.-B. Chang and A. Pires, *Sustainable Solid Waste Management: A System Engineering Approach*. Hoboken, NJ, USA: Wiley-IEEE Press, 2015.
- [18] K. Gu, J. Qiao, and W. Lin, "Recurrent air quality predictor based on meteorology- and pollution-related factors," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 3946–3955, Sep. 2018.
- [19] M. A. Al Mamun, M. A. Hannan, A. Hussain, and H. Basri, "Integrated sensing systems and algorithms for solid waste bin state management automation," *IEEE Sensors J.*, vol. 15, no. 1, pp. 561–567, Jan. 2015, doi: [10.1109/JSEN.2014.2351452](https://doi.org/10.1109/JSEN.2014.2351452).
- [20] T. Liu, B. Tian, Y. Ai, and F.-Y. Wang, "Parallel reinforcement learning-based energy efficiency improvement for a cyber-physical system," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, pp. 617–626, Mar. 2020.
- [21] M. Wan, T. Qu, M. Huang, L. Li, and G. Q. Huang, "Cloud-based product-service systems platform for household solid waste classification management," *IET Collaborative Intell. Manuf.*, vol. 2, no. 2, pp. 66–73, 2020.

- [22] M. Ghahramani, M. C. Zhou, and C. T. Hon, "Toward cloud computing QoS architecture: Analysis of cloud systems and cloud services," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 1, pp. 6–18, Jan. 2017.
- [23] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: [10.1109/JIOT.2016.2579198](https://doi.org/10.1109/JIOT.2016.2579198).
- [24] J. Xia, G. Cheng, D. Guo, and X. Zhou, "A QoE-aware service-enhancement strategy for edge artificial intelligence applications," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9494–9506, Oct. 2020, doi: [10.1109/JIOT.2020.2996422](https://doi.org/10.1109/JIOT.2020.2996422).
- [25] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020, doi: [10.1109/JIOT.2020.2984887](https://doi.org/10.1109/JIOT.2020.2984887).
- [26] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014, doi: [10.1109/JIOT.2014.2306328](https://doi.org/10.1109/JIOT.2014.2306328).
- [27] M. Ghahramani, N. J. Galle, F. Duarte, C. Ratti, and F. Pilla, "Leveraging artificial intelligence to analyze citizens' opinions on urban green space," *City Environ. Interact.*, vol. 10, Apr. 2021, Art. no. 100058.
- [28] M. Ghahramani and F. Pilla, "Leveraging artificial intelligence to analyze the COVID-19 distribution pattern based on socio-economic determinants," *Sustain. Cities Soc.*, vol. 69, Jun. 2021, Art. no. 102848.
- [29] S. Hu, A. O'Hagan, J. Sweeney, and M. Ghahramani, "A spatial machine learning model for analysing customers' lapse behaviour in life insurance," *Ann. Actuarial Sci.*, vol. 15, no. 2, pp. 367–393, 2021.
- [30] A. Hawbani *et al.*, "A novel heuristic data routing for urban vehicular ad hoc networks," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8976–8989, Jun. 2021, doi: [10.1109/JIOT.2021.3055504](https://doi.org/10.1109/JIOT.2021.3055504).
- [31] S.-S. Lee and S. Lee, "Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10450–10464, Oct. 2020, doi: [10.1109/JIOT.2020.2996213](https://doi.org/10.1109/JIOT.2020.2996213).
- [32] T. Ahmad, R. Chai, M. Adnan, and Q. Chen, "Low-complexity heuristic algorithm for power allocation and access mode selection in M2M networks," *IEEE Internet Things J.*, early access, May 11, 2021, doi: [10.1109/JIOT.2021.3079213](https://doi.org/10.1109/JIOT.2021.3079213).
- [33] M. P. Fantì, A. M. Mangini, A. Favenza, and G. Difilippo, "An eco-route planner for heavy duty vehicles," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 1, pp. 37–51, Jan. 2021.
- [34] X. Jin, H. Qin, Z. Zhang, M. Zhou, and J. Wang, "Planning of garbage collection service: An arc-routing problem with time-dependent penalty cost," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2692–2705, May 2021.
- [35] D. Lei, M. Li, and L. Wang, "A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1097–1109, Mar. 2019.
- [36] J. Wy and S. Kim, "The rollon-rolloff waste collection vehicle routing problem with time windows," *Eur. J. Oper. Res.*, vol. 224, no. 3, pp. 466–476, 2013.
- [37] D. Mu, C. Wang, F. Zhao, and J. W. Sutherland, "Solving vehicle routing problem with simultaneous pickup and delivery using parallel simulated annealing algorithm," *Int. J. Shipping Transp. Logist.*, vol. 8, no. 1, pp. 81–106, 2016.
- [38] L. Wang and J. Lu, "A memetic algorithm with competition for the capacitated green vehicle routing problem," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 2, pp. 516–526, Mar. 2019.
- [39] E. B. Tirkolaee, P. Abbasian, M. Soltani, and S. A. Ghaffarian, "Developing an applied algorithm for multi-trip vehicle routing problem with time windows in urban waste collection: A case study," *Waste Manage. Res.*, vol. 37, no. S1, pp. 4–13, 2019.
- [40] E. B. Tirkolaee, M. Alinaghian, M. B. Sasi, and M. M. S. Esfahani, "Solving a robust capacitated arc routing problem using a hybrid simulated annealing algorithm: A waste collection application," *J. Ind. Eng. Manage. Stud.*, vol. 3, no. 1, pp. 61–76, 2016.
- [41] S. Zhang, D. Mu, and C. Wang, "A solution for the full-load collection vehicle routing problem with multiple trips and demands: An application in Beijing," *IEEE Access*, vol. 8, pp. 89381–89394, 2020.
- [42] T. Huang, Y. Gong, S. Kwong, H. Wang, and J. Zhang, "A niching memetic algorithm for multi-solution traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 508–522, Jun. 2020.
- [43] J. Wang, Y. Zhou, Y. Wang, J. Zhang, C. L. P. Chen, and Z. Zheng, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 582–594, Mar. 2016.
- [44] G. Kim, Y. Ong, C. K. Heng, P. S. Tan, and N. A. Zhang, "City vehicle routing problem (city VRP): A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1654–1666, Aug. 2015.
- [45] J. Toutouh, D. Rossit, and S. Nesmachnow, "Computational intelligence for locating garbage accumulation points in urban scenarios," in *Proc. Int. Conf. Learn. Intell. Optim.*, 2019, pp. 411–426.
- [46] L. Amal, L. H. Son, and H. Chabchoub, "SGA: Spatial GIS-based genetic algorithm for route optimization of municipal solid waste collection," *Environ. Sci. Pollut. Res.*, vol. 25, pp. 27569–27582, Jul. 2018.
- [47] D. Guo, L. Zhu, Y. Lu, H. Yu, and S. Wang, "Small object sensitive segmentation of urban street scene with spatial adjacency between object classes," *IEEE Trans. Image Process.*, vol. 28, pp. 2643–2653, 2019, doi: [10.1109/TIP.2018.2888701](https://doi.org/10.1109/TIP.2018.2888701).
- [48] Y. Sun, Q. Jiang, and M. Singhal, "An edge-constrained localized delay-nay graph for geographic routing in mobile ad hoc and sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 4, pp. 479–490, Apr. 2010.
- [49] Z. Wang, M. Zhang, R. Chu, and L. Zhao, "Modeling and planning multimodal transport paths for risk and energy efficiency using AND/OR graphs and discrete ant colony optimization," *IEEE Access*, vol. 8, pp. 132642–132654, 2020, doi: [10.1109/ACCESS.2020.3010376](https://doi.org/10.1109/ACCESS.2020.3010376).



Mohammadhossein Ghahramani (Member, IEEE) received the Ph.D. degree in computer technology and application from Macau University of Science and Technology, Macau, China, in 2018.

He is currently a Research Fellow with University College Dublin, Dublin, Ireland, where he was also a Member of the Insight Centre for Data Analytics. He has over ten peer-reviewed journal papers as a first author. His research interests include smart cities, smart manufacturing, machine learning, artificial intelligence, big data, and IoT.

Dr. Ghahramani has served the community by reviewing more than 100 papers for top journals in the last four years and has been active in organizing international conferences. He is the Co-Chair of IEEE SMCS Technical Committee on AI-based Smart Manufacturing Systems.



MengChu Zhou (Fellow, IEEE) received the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

Then, he joined the New Jersey Institute of Technology, Newark, NJ, USA, where he is currently a Distinguished Professor. He has over 900 publications, including 12 books, more than 600 journal papers (more than 500 in IEEE transactions), 29 patents, and 29 book chapters. His interests are in petri nets, automation, Internet of Things, and big data.

Dr. Zhou is a Fellow of IFAC, AAAS, CAA, and NAI.



Anna Molter is a Postdoctoral Research Fellow with University College Dublin, Dublin, Ireland. Her research is about how the environment affects people's health. A lot of her work has focused on the effects of air pollution on children's health, but she is interested in a wide range of environmental issues, including both urban and rural problems. She likes to use tools, such as apps and storymaps to communicate her findings with the public and to engage them in scientific research.



Francesco Pilla is an Associate Professor of Smart Cities with University College Dublin, Dublin, Ireland. His work lies at the intersection between cities and technologies with the goal to build better cities through technology, innovation and citizen participation. His area of expertise is smart cities and in specific geospatial analysis and modeling of urban dynamics.