Constructive Heuristics for Commercial Waste Collection with Time-Dependent Travel

Rebecca Hamm[1], Ahmed Kheiri[2], and Tim Pigden[3]

[1] STOR-i Centre for Doctoral Training, Lancaster University, UK
r.hamm@lancaster.ac.uk
[2] Department of Management Science, Lancaster University, UK
a.kheiri@lancaster.ac.uk
[3] Optrak Distribution Software Ltd, Ware, UK
tim.pigden@optrak.com

Abstract. Commercial waste collection can be modelled as a vehicle routing problem with a high number of stops per route, corresponding to bins from individual customers. Retail collections may occur in pedestrian precincts, where access is restricted by time of day. Many commercial collections, particularly from retail areas, occur in highly congested zones, such as high streets. Therefore, modelling with time-of-day dependent travel speeds and turning time penalties (e.g., turning right onto a main road) is essential for accurate time estimation. This study aims to investigate heuristics to solve this problem, specifically using a cluster-first, route-second approach for construction heuristics based on graph partitioning of the road network. Problem instances have been generated, and promising results have been achieved.

Keywords: optimisation, heuristics, routing, operations research.

1 Introduction

Commercial waste constitutes 7% of the UK's domestic road freight, with the vast majority transported by internal combustion engine (ICE) vehicles. Inefficient routes can lead to unnecessary congestion, noise pollution and carbon emissions. We present construction heuristics for a waste collection problem that accurately represents a real-word situation. To do this, we have considered a number of characteristics, including managing numerous pickups per round, considering the complexity of handling hundreds of collections per trip, accounting for time-dependent congestion, and adhering to specific time windows, especially for retail collections. As this work has been completed with Optrak[4], we have utilised their knowledge on real-world routing problems as well as models within their framework including staff regulations and time-dependent travel times.

In this work we build upon Optrak's current construction heuristic to produce a heuristic to deal with a realistic scale collection problem. To manage the large

[4] https://optrak.com/

number of visits considered in these problems, we incorporate a graph partitioning algorithm to group similarly located sites. We can then use these groups to route these visits together. Methods of prioritising hard to route/remote visits are considered by inserting them into routes first.

The structure of this paper is as follows: Section 2 presents a review of relevant literature. Section 3 outlines the problem. Section 4 describes the methodology and the generation of problem instances. Section 5 discusses the results obtained, and finally, Section 6 concludes with a summary of findings.

2 Literature Review

In this section, we give an overview of the literature related to the problem considered in this paper.

Buhrkal et al. [4] consider a waste collection Vehicle Routing Problem (VRP) in an urban context, ensuring that all bins are emptied while respecting time windows. Drivers are given breaks as required by law. They used an Adaptive Large Neighbourhood Search (ALNS) method, which has shown positive results on both benchmark instances from the literature and instances provided by a Danish waste company. Similarly, Campos and Arroyo [5] describe a waste collection problem with time windows that includes a lunch break for drivers. They propose a hybridisation of Iterated Local Search (ILS) metaheuristic and Variable Neighbourhood Descent (VND) local search method, which they test on instances provided by Kim et al. [9].

Rabbani et al. [17] describe a waste collection problem using vehicles with separate compartments. Each compartment has a different capacity for various types of waste, and every vehicle must visit all disposal facilities for all waste types. The problem combines the multi-depot vehicle routing problem and the mixed close-open VRP, involving both internal vehicles that must return to the depot and external vehicles that do not. A hybrid genetic algorithm incorporating an iterated swap procedure to improve solutions is tested on randomly generated instances.

Babaee Tirkolaee et al. [3] present a waste collection with time windows. The problem also has a maximum time allowance for vehicles. Simulated annealing is used to produce solutions for randomly generated instances. Masmoudi et al. [14] describe a waste collection problem that uses plug-in hybrid vehicles powered by two sources: electricity and compressed natural gas. The problem considers both refuelling and recharging. They propose a hybrid acceptance threshold algorithm, which is tested on a set of adapted well-known EVRP (Electric VRP) benchmark instances. Erdem [6] presents a waste collection problem using a heterogeneous fleet of electric vehicles, considering multiple types of waste. These waste types are organised into multi-compartments and are picked up within specified time windows. The problem also addresses split deliveries and waste bin-vehicle compatibility. An adaptive variable neighbourhood search (AVNS) algorithm has been developed to efficiently solve a generated set of instances. Yang et al. [19] describe an electric vehicle waste collection problem that considers a chance-constrained collection and transportation problem for sorted waste using multi-compartment electric vehicles. A chance constraint on the amount of waste generated is used to manage the unpredictability of waste production. They propose a diversity-enhanced particle swarm optimisation with neighbourhood search (DNSPSO) and particle swarm optimisation with simulated annealing (PSO-SA), collectively called DNSPSOSA, which are tested on a benchmark set of instances.

Qiao et al. [16] consider a waste collection problem where vehicles can take multiple trips. Several objectives are considered: fixed vehicle costs, fuel consumption, carbon emissions, and penalty costs. Penalty costs are imposed when the number of vehicles assigned to a disposal facility surpasses its processing capacity. A metaheuristic combining particle swarm optimisation and tabu search is applied to a set of small-scale instances. Akbarpour et al. [1] formulate two problems: the first is a VRP to a waste separation site, and the second is resource allocation from the waste separation site to recovery plants or landfill centres. In this problem, the city is divided into different sections or areas, each with a separation centre serving as a separation unit. A Genetic Algorithm (GA) and Simulated Annealing (SA) are compared, along with two hybrid algorithms: GA-SA and GA-PSO.

Malandraki and Daskin [13] describe a time-dependent VRP where the travel time between two points depends on both the distance between these nodes and the time of day. A nearest neighbour heuristic is used to solve the problem. Time-dependent travel times are generally modelled following the example of Ichoua et al. [8], where the workday is partitioned into several periods, and a constant travel speed is assigned to each time interval. This results in the speed being a step function of the departure time for all arcs.

Kim et al. [9] present a waste collection problem with time windows, incorporating multiple disposal trips and drivers' lunch breaks. Multiple objectives are considered: minimising the number of vehicles, minimising travel time, maximising route compactness, and balancing workload. An extended insertion algorithm and a clustering-based algorithm are used to solve the problem. Mat et al. [15] use a current initial solution algorithm and a different initial customer algorithm for a time-dependent waste collection problem, considering dynamic vehicle speeds (the heavier a vehicle, the slower it is).

Gómez et al. [7] study a real-world instance raised by local authorities in a rural region of northwestern Spain. They address a bi-objective problem over a monthly planning period, considering total costs and service level. Service quality deteriorates as waste accumulation increases. They impose a maximum allowed separation time between collections at the same point. To solve this, they employ Multiobjective Adaptive Memory Programming using tabu search to identify non-dominated solutions and approximate the efficient frontier.

Aliahmadi et al. [2] propose a waste collection problem with multiple depots and intermediate facilities, which are utilised in developing countries to reduce long-term costs and improve the quality of waste collection processes. They employ a fuzzy optimisation approach to handle uncertainty in waste amounts.

A GA is implemented using real data from the Tehran district in Iran, and solutions are validated through comparison with the current real-life situation.

Objectives, model features, and the solution methods used are summarised for each paper in Table 1. For a comprehensive overview, interested readers may refer to Liang et al. [11] for a recent survey.

Table 1: Summary of objectives, model features, and solution methods in selected waste collection optimisation studies

		Model Features									
Paper	Objective	Split delivery	Time window	Empty return to depot	Time dependency	Load capacity	Landfill trips as needed	Heterogeneous fleet	Maximum time	Staff breaks	Method of solution
[14]	Traversing costs		*	*		*	*		*		Hybrid acceptance threshold
[3]	Vehicle, traversing and time window penalty costs		*			*	*	*	*		SA
[6]	Travel distance		*			*		*	*		AVNS
[19]	Fixed and energy costs of EVs					*					DNSPOSA
[16]	Vehicle, fuel, CO_2 and penalty costs			*		*	*				PSO-TS
[1]	Travel distance		*			*	*	*			Hybrid SA, GA and PSO
[9]	Vehicle, travel time, route compactness and balanced workload		*	*		*	*		*	*	Clustering-based algorithm
[17]	Total cost			*		*	*	*	*		Hybrid GA
[5]	Total cost		*	*		*	*			*	ILS-VND
[13]	Minimum time		*		*	*		*			Nearest neighbour
[4]	Cost of arcs		*			*	*			*	ALNS
[7]	Cost and quality of service	*		*		*	*	*			MOAMP
[2]	Total cost			*		*	*	*			GA

3 Problem Description

We describe a vehicle routing problem that involves collecting waste from a large number of commercial sites. To accurately represent the real-world scenario, we consider time-dependent travel times. These travel times are calculated using piecewise linear functions to account for fluctuations throughout the day due to factors such as changes in congestion.

Limitations on staff shifts are enforced to coincide with legal requirements. This includes compulsory staff breaks based on both the total working time and the amount of time spent driving. A maximum shift length is also considered enforcing a time constraint on the routes within the solution.

Due to council restrictions as well as high levels of congestion at certain times of day on high streets and pedestrian precincts, we have time windows on collection points within these areas.

Each collection site can only be visited once. All vehicles have a limited capacity of waste which it can transport. When a truck is full, it must visit a landfill site to unload before we can complete any further collections. We consider large instances to accurately represent the problem. However, this means routing to all sites is not always possible. Therefore, we do not enforce a constraint to satisfy all demand. Instead, we include a penalty cost for unrouted collections within our overall costs. Our objective is to minimise the total costs. This includes costs from number of vehicles, distance travelled, staff costs as well as penalty costs. The problem is described below:

Objective

Minimise total costs including vehicle, travel, staff, and penalty costs.

Constraints

- Each bin can only be visited once.
- Bins must be collected within specified time windows.
- Vehicles have a limited waste capacity.
- Vehicles must visit a landfill site when full.
- Worker breaks and driving times must comply with legal requirements.

4 Methodology

4.1 Test Instances

We have looked at three different areas when creating instances: Lancaster and surrounding areas, East Hertfordshire and Central Leeds. This is to provide a range of geographical features. We have sourced information of commercial properties for these areas from OS AddressBase[5] and Foursquare API[6].

The sites found from these sources have been combined and sampled. We generated instances of various sizes to simulate companies having different proportions of the market.

We simulated bin data using real (WRAP) bin data, restricting bin types to just two sizes: 240-litre and 1100-litre bins. Bins for each site were simulated based on waste type and business type, ensuring that the quantity of bins collected from a site accurately reflects what that company may produce.

We consider different waste types: residual, glass, and food. Details of the instances produced are shown in Table 2.

[5] https://www.ordnancesurvey.co.uk/products/addressbase-premium
[6] https://location.foursquare.com/developer/

Table 2: Details of instances produced, including location, number of customers, number of vehicles, and waste type

Instances	**Location**	**Customers**	**Vehicles**	**Type**
Instance 1	East Herts	2000	19	food
Instance 2	East Herts	2000	19	glass
Instance 3	East Herts	3500	33	residual
Instance 4	East Herts	3500	33	food
Instance 5	Lancs	2500	23	residual
Instance 6	Lancs	3000	28	residual
Instance 7	Lancs	3000	28	glass
Instance 8	Leeds	2000	19	glass
Instance 9	Leeds	5000	47	residual
Instance 10	Leeds	5000	47	food

Using QGIS software[7], we identified areas containing high streets. For these areas we have enforced time windows to prevent driving through the areas in heavy congestion times. These time windows are: 06:00AM-08:00AM, 10:00AM-15:00PM, and 18:00PM-20:00PM.

Time dependencies and staff breaks have been implemented within Optrak's model. To simulate a landfill trip within the journey, we have doubled the vehicle's capacity and reduced the maximum journey time by two hours.

4.2 Heuristics for Commercial Waste Collection

We employ construction heuristics for these waste collections, advancing beyond the basic heuristic currently used by Optrak. Specifically, we utilise graph partitioning methods to group similarly located sites and optimise their routing together.

Original Heuristic (OH) The original heuristic is a modified version of Mole and Jameson's sequential route building algorithm [10].

The heuristic considers one vehicle at a time. Initially, a seed source is used to select the first visit to be placed into this vehicle. Selection of subsequent visits uses a roulette wheel method, where probabilities are determined by a parameter. The remaining unrouted visits are organised into a heap, ordered by proximity to the non-depot visits of the new trip.

Next, the heuristic iteratively selects the top visit from the heap and attempts to insert it at the best position within the route. If successful, the heap is updated and the visit is removed. If insertion fails, the visit remains in the heap. This process continues until either the heap is exhausted (all visits are routed) or visits can no longer be placed in the vehicle without breaking constraints. The procedure then repeats for the next vehicle in the sequence.

[7] https://qgis.org/en/site/

Once all vehicles have been assigned routes, we apply the Lin-Kernighan heuristic [12] with depth 2 to refine the routes. This heuristic operates on one vehicle's route at a time, iteratively selecting and removing edges of a route, and replacing them with cheaper edges so that the route still starts and ends at the depot, and has a lower overall cost.

Graph Partition Based Heuristic (GP) Due to the scale of our problem instances, we adapt the previous heuristic by using graph partitioning to create groups of customers which are in close proximity to each other and hence can be expected to be in close proximity of each other in a route.

We use KaFFPa [18] to create these partitions. This is a successful heuristic for partitioning large graphs which recursively contracts a graph to create smaller graphs. After applying an initial partitioning algorithm to the smallest graph, the contraction is undone and, at each level, a local search method is used to improve the partitioning induced by the coarser level.

At the start of the algorithm, we apply the graph partitioning algorithm to our customer base to create groups of visits. Similar to the original heuristic, we consider one vehicle at a time and select the first group of visits using a roulette wheel method, where probabilities are based on a parameter. The closest visit to any visits already in the route is selected for insertion into the route. However, in this heuristic, the entire partition containing that visit is inserted together in the best feasible position.

Within the partition, the visit closest to the visit before the insertion position is placed at the start of the partition, and the visit closest to the visit after the insertion position is placed at the end. These are inserted into the route first, followed by the remaining visits being inserted between them. As before, once visits can no longer be placed in the vehicle without violating constraints, the next vehicle is considered in the same manner. If a vehicle can no longer place visits in its route mid-partition, the routed visits remain routed and the rest of the partition is still considered for later insertion in the heuristic.

Finally, as before, we apply the Lin-Kernighan heuristic [12] with depth 2 to the completed routes to further optimise them.

Algorithm 1 provides the pseudocode for the graph partition based heuristic. Figure 1 shows the partitions applied to instances 1 and 9.

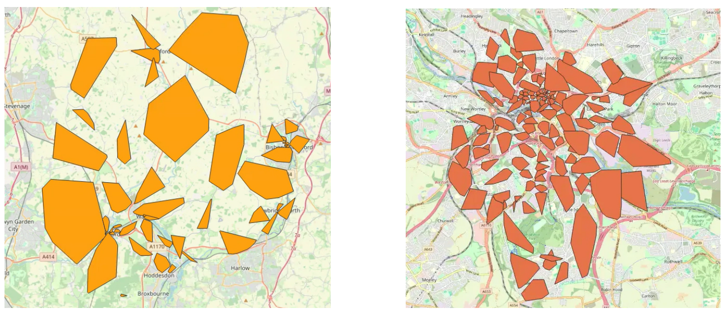
Seed Source Parameters We consider four different seed source parameters. The first two, "min" and "long", are used for both heuristics, while "far" and "two" are specifically designed for the graph partitioning algorithm.

When applied within the Graph Partition Based Heuristic, these seed source parameters select a visit first, followed by the selection of the partition containing that visit.

- **Min:** This parameter finds the shortest distance to another site for each site and favours the site with the smallest distance.

Algorithm 1: Graph Partition Based Heuristic

```
Input: Problem(list of sites, demand, vehicles)
Output: routes
Partitions =GraphPartition(Sites);
foreach vehicle do
    FirstPartition= SeedSource(Partitons);
    Insert(FirstPartition);
    while visits can feasibly be inserted do
        NextPartition= SelectClosest(Partitions);
        Insert(NextPartitionn)
    end while
end foreach
return routes
```



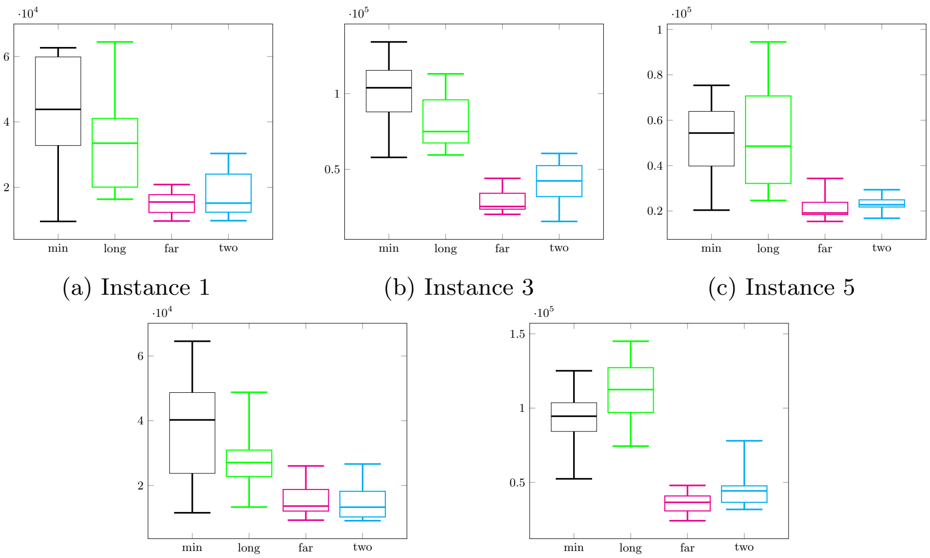
(a) Instance 1 (b) Instance 9

Fig. 1: Partitions shown on instances 1 and 9

- **Long:** This parameter finds the longest distance to another site for each site and favours the site with the smallest distance.
- **Far:** This parameter is used to prioritising routing groups of visits in remote locations. For each group of visits, their connections are ordered based on distance and we consider the smallest distance. We favour the group of visits with the largest smallest distance. These are the groups of visits where the closest neighbours are the furthest away, hence we can assume they are in more remote locations.
- **Two:** Similarly to “Far”, this parameter prioritises groups of visits where the closest neighbours are the furthest away. However, it considers the two closest neighbours to account for cases where one group of visits has a closest neighbour that is further away but many neighbours at a similar distance, whereas another group has a closer nearest neighbour but the next closest is extremely far away.

5 Results

We aim to determine which seed source parameter works best with our Graph Partition Based Algorithm. We ran the algorithm 10 times using each parameter on five of our instances, chosen to represent a range of different sizes, locations, and product types. The results from this experiment are reported in Figure 2.



(a) Instance 1 (b) Instance 3 (c) Instance 5

(d) Instance 8 (e) Instance 9

Fig. 2: Comparison of seed source parameters

For instances 3 and 9 far and two are outperforming min and long on the majority of their runs so are favourable from these instances. We can see for instances 1, 5 and 8 that the costs from min and long span a large range which overlaps with far and two however, the costs from parameters far and two have a smaller range over the lower costs. Hence far and two appear to be performing better on these instances.

It is clear from these results that starting with remote locations when building a route aides the algorithm in finding a good solution. For Instances 3, 5 and 9 the median and upper quartile are less for far than two suggesting the majority of runs have solutions with lower costs. For instance 1, far has a slightly higher median than two but all runs have a lower cost than the upper quartile of the costs from using two. Results from using far and two appear similar for instance 8.

From these results, we have concluded that far appears to be working best, hence we will compare the results of our heuristic using far against the original heuristic.

To compare our heuristic with the original heuristic, we ran both algorithms 10 times on all 10 instances. Since the far parameter is specifically designed for the Graph Partition Based Algorithm and not suitable for the original heuristic, we used long as the parameter for the original heuristic. This choice is based on previous performance knowledge and our results from the earlier experiment. The results of this comparison are presented in Table 3.

Table 3: Average results of running GP and OH algorithms 10 times on 10 instances. Metrics include the average cost (avg), standard deviation (std), best cost (best), and average number of unrouted sites (unr)

Instance	GP Heuristic				OH Heuristic			
	avg	std	best	unr	avg	std	best	unr
Instance 1	**15105.68**	3926.22	9702.04	45.8	15967.77	8115.67	**9157.31**	66.6
Instance 2	9771.88	1512.64	7667.14	17.9	**8312.78**	388.59	**7573.96**	4.3
Instance 3	**28904.75**	7601.92	**20267.09**	116.1	81538.45	17028.32	47795.47	544.1
Instance 4	19993.08	3040.12	16397.69	58.6	**14418.02**	745.75	**13114.90**	11.08
Instance 5	**21294.70**	5618.46	**15416.95**	92.9	55710.61	11149.82	39887.45	405
Instance 6	15307.61	4080.32	10840.77	45.4	**11696.51**	1122.29	**10364.65**	13.6
Instance 7	**45752.76**	10066.60	34020.59	281.7	55368.86	22281.45	**20961.79**	383.4
Instance 8	15412.50	5336.81	9290.67	76.5	**10752.74**	2984.987	**6011.88**	46.7
Instance 9	**35910.56**	7597.13	**24196.96**	183	53803.63	19171.49	24772.15	346.9
Instance 10	23593.10	3540.88	18725.14	86.9	**15304.73**	371.23	**14765.77**	12

Our results indicate that the performance of the heuristics varies across instances. However, we observe significant improvements with the Graph Partition Based Heuristic for instances 3 and 5, reducing the mean cost by 65% and 61%, respectively. Furthermore, the Graph Partition Based Heuristic outperforms the Original Heuristic on instances with a larger number of difficult-to-route visits. This makes it a favourable choice for instances requiring more intricate routing solutions.

6 Conclusions

We have addressed a commercial waste problem incorporating time-dependent travel times, time windows, and staff breaks. To accurately simulate this problem, we compiled data from various real-world sources. Utilising graph partitioning, we enhanced an existing construction heuristic to insert groups of visits at a time on certain problem instances.

Exploring different methods to select initial visits for insertion into routes, we conducted experiments to determine the most effective approach. Our comparative analysis against the existing heuristic revealed that our new approach outperforms on half of our instances. Specifically, it excels on instances where the existing heuristic struggles with routing a larger number of visits.

We hypothesise that our construction heuristic is particularly advantageous for instances with visits that pose greater routing challenges. However, it remains unclear whether this improvement is attributable to the graph partitioning technique or the "far" parameter, which prioritises routing remote visits first.

Acknowledgement

We gratefully acknowledge financial support from the Innovate UK KTN under the Accelerated Knowledge Transfer (AKT) scheme, and from Optrak Distribution Software Ltd (Project ID 1860)

References

[1] Navid Akbarpour, Amirhossein Salehi-Amiri, Mostafa Hajiaghaei-Keshteli, and Diego Oliva. An innovative waste management system in a smart city under stochastic optimization using vehicle routing problem. *Soft Computing*, 25:6707–6727, 2021.

[2] Seyed Zeinab Aliahmadi, Farnaz Barzinpour, and Mir Saman Pishvaee. A fuzzy optimization approach to the capacitated node-routing problem for municipal solid waste collection with multiple tours: A case study. *Waste Management & Research*, 38(3):279–290, 2020.

[3] Erfan Babaee Tirkolaee, Parvin Abbasian, Mehdi Soltani, and Seyed Ali Ghaffarian. Developing an applied algorithm for multi-trip vehicle routing problem with time windows in urban waste collection: A case study. *Waste Management & Research*, 37(1_suppl):4–13, 2019.

[4] Katja Buhrkal, Allan Larsen, and Stefan Ropke. The waste collection vehicle routing problem with time windows in a city logistics context. *Procedia-Social and Behavioral Sciences*, 39:241–254, 2012.

[5] Alba A Campos and José Elias C Arroyo. An ils heuristic for the waste collection vehicle routing problem with time windows. In *Intelligent Systems Design and Applications: 16th International Conference on Intelligent Systems Design and Applications (ISDA 2016) held in Porto, Portugal, December 16-18, 2016*, pages 889–899. Springer, 2017.

[6] Mehmet Erdem. Optimisation of sustainable urban recycling waste collection and routing with heterogeneous electric vehicles. *Sustainable Cities and Society*, 80:103785, 2022.

[7] José Rubén Gómez, Joaquín Pacheco, and Hernán Gonzalo-Orden. A tabu search method for a bi-objective urban waste collection problem. *Computer-Aided Civil and Infrastructure Engineering*, 30(1):36–53, 2015.

[8] Soumia Ichoua, Michel Gendreau, and Jean-Yves Potvin. Vehicle dispatching with time-dependent travel times. *European journal of operational research*, 144(2):379–396, 2003.

[9] Byung-In Kim, Seongbae Kim, and Surya Sahoo. Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33 (12):3624–3642, 2006.

[10] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, page 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603638.

[11] Yun-Chia Liang, Vanny Minanda, and Aldy Gunawan. Waste collection routing problem: A mini-review of recent heuristic approaches and applications. *Waste Management & Research*, 40(5):519–537, 2022.

[12] Shen Lin and Brian W Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973.

[13] Chryssi Malandraki and Mark S Daskin. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation science*, 26(3):185–200, 1992.

[14] M Amine Masmoudi, Leandro C Coelho, and Emrah Demir. Plug-in hybrid electric refuse vehicle routing problem for waste collection. *Transportation Research Part E: Logistics and Transportation Review*, 166:102875, 2022.

[15] Nur Azriati Mat, Aida Mauziah Benjamin, and Syariza Abdul-Rahman. Enhanced heuristic algorithms with a vehicle travel speed model for time-dependent vehicle routing: A waste collection problem. *Journal of Information and Communication Technology*, 17(1):55–78, 2018.

[16] Qingqing Qiao, Fengming Tao, Hailin Wu, Xuewei Yu, and Mengjun Zhang. Optimization of a capacitated vehicle routing problem for sustainable municipal solid waste collection management using the pso-ts algorithm. *International journal of environmental research and public health*, 17(6):2163, 2020.

[17] Masoud Rabbani, Hamed Farrokhi-Asl, and Hamed Rafiei. A hybrid genetic algorithm for waste collection problem by heterogeneous fleet of vehicles with multiple separated compartments. *Journal of Intelligent & Fuzzy Systems*, 30(3):1817–1830, 2016.

[18] Peter Sanders and Christian Schulz. Engineering multilevel graph partitioning algorithms. In *European Symposium on algorithms*, pages 469–480. Springer, 2011.

[19] Jia Yang, Fengming Tao, and Yanni Zhong. Dynamic routing for waste collection and transportation with multi-compartment electric vehicle using smart waste bins. *Waste Management & Research*, 40(8):1199–1211, 2022.