# Improving Bus Schedules and Waste Collection Routes in Practical Smart City Implementations

Jan Dünnweber
Ostbayerische Technische Hochschule Regensburg
jan.duennweber@othr.de

Amitrajit Sarkar
Ara Institute of Canterbury
amit.sarkar@ara.ac.nz

Vimal Kumar Puthiyadath
KPIT Technologies Ltd
Vimal.Puthiyadath@kpit.com

Omkar Barde
KPIT Technologies Ltd
Omkar.Barde@kpit.com

## Abstract

*Computer-based Improvements of waste collection and public transport procedures are often a part of smart city initiatives. When we envision an ideal bus, it will primarily connect the most crowded bus stops. Similarly, an ideal waste collection vehicle will arrive at every container exactly at the time when it is fully loaded. Beyond doubt, this will reduce traffic and support environmentally friendly intentions like an expansion of waste separation, as it will make more containers manageable. An obvious difficulty of putting that vision into practice is that vehicles cannot always be where they are needed. Knowing the best time for arriving at a certain position is insufficient for finding the optimal route. Therefore, we compare four different approaches to optimized routing: Regensburg, Christchurch and Bangalore and Malaysia. Our analysis shows that the most efficient schedules result from adapting field-tested routes frequently on the basis of sensor measurements and shortcuts resulting from route optimization computations.*

## 1. Introduction

Even with the latest IoT technology like networked sensors and simulations forecasting the behaviour of the population of megacities, practical implementations of on-demand public transport and smart waste collection still have difficulties in keeping up with the prognosticated improvements. A vehicle that drives obstinately from the most crowded bus stop (or the most heavily filled container, respectively) to one with the fill-grade closest to that will obviously need more time in the majority of cases than a vehicle following a fixed plan, since the crowded bus stops and the heavily filled containers are probably positioned far apart from each other.

Finding a smarter route leads us to the classic *vehicle routing problem* (VRP [1]), an instance of the *Travelling Salesman Problem (TSP)* with the added constraint that we need to return to the starting point after visiting a fixed number of points, since the collection vehicle has a limited capacity. Thus, we don't need to find the minimum Hamiltonian circle through all the points but multiple circles forming some kind of clover leaf. However, finding the best route does not only require to consider the distances between the single stops. Busses should leave out stops where no passengers are waiting and waste collectors should skip containers which are only to a certain level, i. e. we are dealing with an instance of a dynamic route planning problem, which is also the subject of more recent research [2].

There are $\frac{n!}{2}$ different routes connecting $n$ positions. For comparing all possible routes between only $10$ stops, this means 3628800 routes must be analyzed. While a typical bus line might comprise $10$ stops, a modern waste collection vehicle can be loaded with $\approx$ 400 containers of 120 liters [3]. $400!$ is a 882-digit number. Taken into account that skipping containers with little load means, the vehicle has to pickup an other one, where it usually does not drive to, solving our dynamic VRP requires to solve a new problem of that size, every time when the fill-level measurements are updated. Nowadays, supercomputers can deal with such problem sizes [4]. However, the presented projects deal with approximate solutions, which can be found using a standard PC or an on-board computer in a bus or garbage truck. Therefore, the presented work might be relevant for automating the navigation of future self-driving busses and garbage trucks, like the one Volvo started testing in Brussels recently [5].

The rest of this paper is structured as follows: Section 2 explains the *ant colony optimization* (ACO) and shows how Regensburg and Christchurch benefit from ACO and fill-level sensing for collecting their waste containers. Section 3 introduces a different approach to minimizing routes using a greedy algorithm and shows two more practical implementations: The waste collection in Malaysia and the employee transport service of Bangalore. Section 4 discusses the *lessons learned* and points out some future perspectives.

## 2. Routing by means of the Ant Colony Optimization

The *ant colony optimization* is an approximate solution for TSP instances (ACO [6]). This approach has been proven suitable for dynamic VRP instances in a simulation, where the road network and the related traffic were taken into account [7]. ACO is a *swarm intelligence* procedure, i. e. not an individual (a simulated ant in the case of ACO) solves a problem but a group. For finding optimal routes, the simulation starts with letting the ants take random paths until they reach their destination. This random walk is optimized iteratively: each ant leaves a pheromone trail behind it which evaporates after a certain number of iterations. In every iteration the pheromone intensity of the shorter paths increases because whenever a simulated ant can choose among multiple paths, it takes the one with the highest pheromone intensity. This means, in higher iterations, the paths are no more randomly chosen but influenced by the most successful ants from preceding iterations, which are the ones whose pheromone trails did not evaporate until their followers reached them, since they were on the shortest paths.

### 2.1. Application 1: The Collection of Biological Waste in Regensburg

Our work in Regensburg focuses on improving the collection of biological waste [8]. While other approaches toward the computer-aided routing of waste collection vehicles rely only on fill-level sensing [9] or only on ACO-based path computations [10], we combined both ideas.

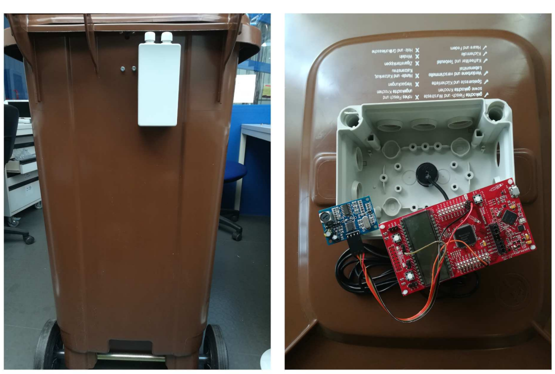

Figure 1. Container (left) and electronics (right)

Simulations have shown, that the combination of these techniques can significantly reduce the time needs for the collection of waste [11]. Since the present contracts with service providers or logistical obstacles often conflict with changes to the waste collection routes, possible time savings could only be computed but not really be exploited in many places.

Contrarily, in Regensburg, we recently started applying route optimization and fill-level sensing to the collection of the containers for biological waste. Figure 1 (left) shows one such container and the protection casing on the back which houses our electronics. Actually, the right part of the picture with the detail view of the electronics shows an earlier prototype where the electrical parts placed in the lid. The new box is smaller (making it more difficult to recognize single parts like our LoRa transmitter) and delivers correct data, even if the lid of the container is not closed.

Regensburg established a novel program for the collection of biological waste in 2018 and equipped the city and its surrounds with 700 new containers. The novelty of the program allows us, computer scientists (headed by author Jan Dünnweber) and electrical engineers (headed by his colleague Martin Schubert) at the *Technical University of Applied Sciences* in Regensburg (OTH), to contribute to the implementation. Noteworthy as well, is that 700 containers are not too many. While the number of possible thorough paths, connecting all containers is tremendous ($\frac{n!}{2}$ is a 1719-digit number), there are only $\binom{700}{2}$ ways to choose an interconnecting route between an unordered pair of containers, which is $\frac{700!}{2!(700-2)!} = \frac{700!}{2\times(698)!} = \frac{699\times 700}{2} = 244650$ interconnections. Combinations vary, but the interconnections are static and can be stored. Therefore, we set up a database, which holds for half of the 700 containers (*one-way*) a file with the distances connecting it to the remaining 699 candidates (244650 entries totally).

| Source | Destination | Distancein Km |
|---|---|---|
| Cranachweg 18, 93051 Regensburg, Germany | Dr.-Gessler-Straße 47, 93051 Regensburg, Germany | 2.4 km |
| Cranachweg 18, 93051 Regensburg, Germany | Dr.-Gessler-Straße 22, 93051 Regensburg, Germany | 0.4 km |
| Cranachweg 18, 93051 Regensburg, Germany | Dr.-Gessler-Straße 22, 93051 Regensburg, Germany | 0.4 km |
| Cranachweg 18, 93051 Regensburg, Germany | Dr.-Gessler-Straße 47, 93051 Regensburg, Germany | 2.4 km |
| Cranachweg 18, 93051 Regensburg, Germany | Defreggerweg 4, 93051 Regensburg, Germany | 0.7 km |
| Cranachweg 18, 93051 Regensburg, Germany | Kirchmeierstraße 20, 93051 Regensburg, Germany | 1.0 km |
| Cranachweg 18, 93051 Regensburg, Germany | Kirchmeierstraße 22, 93051 Regensburg, Germany | 0.7 km |
| Cranachweg 18, 93051 Regensburg, Germany | Simon-Sorg-Straße, 93051 Regensburg, Germany | 0.8 km |
| Cranachweg 18, 93051 Regensburg, Germany | Simon-Sorg-Straße, 93051 Regensburg, Germany | 0.8 km |
| Cranachweg 18, 93051 Regensburg, Germany | A93, 93049 Regensburg, Germany | 3.2 km |
| Cranachweg 18, 93051 Regensburg, Germany | Georg-Herbst-Straße 35, 93049 Regensburg, Germany | 1.6 km |
| Cranachweg 18, 93051 Regensburg, Germany | Georg-Herbst-Straße 18, 93049 Regensburg, Germany | 1.7 km |
| Cranachweg 18, 93051 Regensburg, Germany | Georg-Herbst-Straße 18, 93049 Regensburg, Germany | 1.7 km |
| Cranachweg 18, 93051 Regensburg, Germany | Clausewitzstraße 14, 93049 Regensburg, Germany | 1.9 km |
| Cranachweg 18, 93051 Regensburg, Germany | Scharnhorststraße 24, 93049 Regensburg, Germany | 1.8 km |
| Cranachweg 18, 93051 Regensburg, Germany | A93, Regensburg, Germany | 3.6 km |
| Cranachweg 18, 93051 Regensburg, Germany | Lilienthalstraße 1, 93049 Regensburg, Germany | 2.6 km |
| Cranachweg 18, 93051 Regensburg, Germany | Prüfeninger Str. 141, 93049 Regensburg, Germany | 2.6 km |

Figure 2. Distances in an Excel sheet

To ascertain the profitability of our undertakings, we also started with a simulation. However, our simulation did not forecast the profits of ACO on arbitrary routes or the benefits from observing arbitrary containers. We computed a viable forecast for Regensburg. With the help of a group of students (David Burger, Vadim Dechand, Haris Shehzad and Markus Wildgruber), we could fill the mentioned 244650-entry database with concrete distances and average driving times. Figure 2 shows the first 19 entries of this database in an Excel-sheet. Actually, we maintain this data in Redis to benefit from caching when the same distances are requested repeatedly [12].

After accompanying the waste collectors and recording the GPS positions of the containers and the time needs for emptying them with a fitness tracking app on the smart phone, we requested the distances for all possible pairs from the *Google Maps* Web service by means of a Java program, which the students have developed to export the distance data into the popular TSPLIB-format [13]. With this representation, our data can be processed using Open-Source ACO-code and other TSP-solvers. Our routing software makes use of the Thomas Stützle Implementation [6] and we used the exact TSP-solver *Concorde* [14] as a reference. Figure 3 shows the route we computed for connecting 11 containers in red. This route is 11 kilometers long and could be shortened to 7 kilometers (green route) after leaving out two containers, which we identified as empty containers using our fill-level sensors.

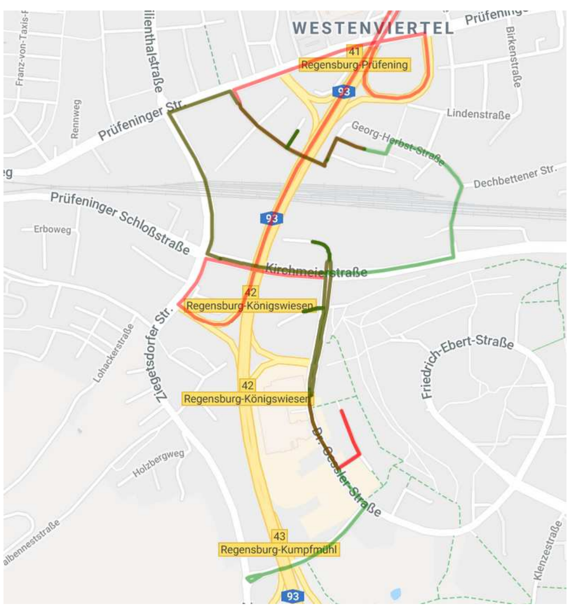


**Figure 3. Container (left) and electronics (right)**

To integrate fill-level sensing into the route computations, we started to cope with underfilled containers. While overfilled containers seem to be a more urgent problem, it is difficult (or almost impossible) to send a vehicle instantaneously, when an overfilled container is detected. Planning collection routes such that underfilled containers are skipped is much easier and saves time. That saved time is used to collect new containers. The recorded data about overfilled containers helps to position the new containers where they are needed. Thus, our waste management software, tackles both problems, underfilled containers directly (by leaving them our during the collection) and overfilled containers as well, by finding the best positions for new containers in the long run. Instead of resetting the route computations when a container is added, removed or re-positioned, we let ACO continuously running, since it is known that the swarm algorithm can adapts to its input [15].

Master-student Josef Weiß (from Martin Schubert's group) set up the electronics shown in Figure **??**: The larger board (on the right) holds an ultrasonic transceiver module and a low power C. The smaller board (on the left holds) and a LoRa (Low Range) transceiver. We communicate the fill grade measured by the ultrasonic transceiver via a LoRa gateway. The costs for 10 of this *DIY*-devices were below 1000 € and sponsored by `kpit.com`.

Using our simulation, we could predict that 10 sensors are enough to start benefiting from our software. To compute a trustworthy prediction, our simulation does not simply roll the dice to decide whether a sensor-equipped container can be skipped. Instead the probability for each container $i$ to be empty was computed individually using the formula: $P(x_i) = q_i * \frac{c_i}{d_i}$. The value for $q_i$ ranges between $0.1$ (overfull), $0.2$ (full), $0.3$ (half-full) and $0.5$ (empty) and was set accordingly to the average fill level of that container that we observed on the collection tours. Parameter $d_i$ takes account for the fact that waste containers located near the city center are more likely found full and is set to $1.2$ for a container withing a $2$ kilometer circle around the center, $1.1$ withing $4$ kilometers and $0.9$ for containers located more than $6$ kilometers away from the center. Parameter $c_i$ is weighted accordingly to the number of waste containers next to it and set to $0.9$ when there is none or set to $1.1$ for up to $5$ neighboring containers or set to $1.3$ when there are $6$ or more. With that estimations, we predicted time savings of approximately one hour per day. Regensburg has just started to adapt the routes accordingly. Thus, we will soon know how many extra containers can be emptied within the saved time.

### 2.2. Application 2: LevelSense™ and the PiP-IoT project

While New Zealand may be considered a small country, its smart city ambitions are anything but small.

For the last two consecutive years New Zealand earned more recognitions for their smart cities initiatives than any other country in the annual Smart City Asia Pacific Awards (SCAPA), a competitive benchmarking program conducted by Council Associate Partner IDC. Following the significant earthquakes of 2010/11, Christchurch City Council (CCC) put aside funds for Sensing Cities initiatives. In November 2015, the Smart Cities programme was initiated to carry out rapid proof of concept projects.

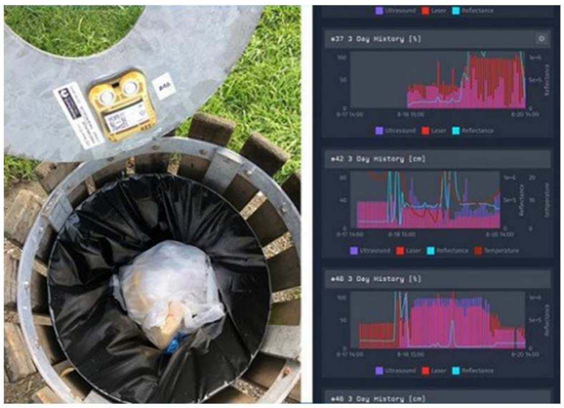

**Figure 4. LevelSense™ in container lid (left) and the Christchurch online monitoring app (right)**

More than 50% of the worlds population lives in cities by 2050, it will be more the 75%. While the worlds cities only cover two per cent of the global land area, they are responsible for 70% of greenhouse gas emissions and the global climate change. As a consequence, we urgently need sustainable solutions to combat these city-focused issues. Smart garbage collection, energy-saving and efficient street and traffic lighting, water and wastewater management, and the reduction of $CO_2$ emissions from motor vehicles are all essential elements of these solutions. Christchurch is using smart sensors to ensure rubbish bins are emptied at the right time.

Overflowing public rubbish bins are a frequent source of complaints from residents. Citys rubbish collection contractors work constantly to empty the citys bins but before they arrive at the site they don't know whether theyll find an empty bin, a full one or an overflowing mess. The bin sensor trial uses LevelSense™ sensors, which were developed by Christchurch company PiP IoT, to check rubbish levels. The sensors allow city councils contractors to recognise optimal waste collection times and recommend optimal collection routes. As a result, the council's operational managers can find out whether they have the right number of bins in the right places to meet public demand.

## 2.3. Level Sensor Specifications

Level Sense sensors where built for ultra-rugged tank and bin level measurement with GPS position and movement monitoring. The devices monitor the fill levels of remote tanks, bins or other container assets in real time for improved service management outcomes.

The key features are

- Tilt sensing capability
- GPS location and Geo-fencing capabilities
- Tampering and orientation detection
- Shock and vibration sensing
- Temperature measurement
- Expected battery life of 5 years
- Low Power Network - LPWAN Connectivity
- Real time analytics
- Mobile / Web Reporting
- SIGFOX Class 0 Certified

**2.3.1. How the Sensors Work** A small unit mounted on the inside of the bin lid uses sonar, which detects the rubbish level and transmits readings to the monitoring system (see figure 4, left).

A dashboard shows the status of each bin. When rubbish reaches a specified level, the system alerts the contractor that the bin needs emptying.

### 2.3.2. Anticipated Outcomes

- Reduction in CO2 emissions and pollution: fewer rubbish collection trucks on the road for less time, which means lower fuel consumption and lower greenhouse gas emissions. Fewer collection trucks on the road will also mean less noise pollution, air pollution, and less wear and tear on our road network.
- Reduction in operational costs: managing waste takes a large portion of our rates. Bin level sensors and monitoring solutions have been known to reduce waste collection costs by up to 50 % (fewer collections mean less money spent on driver hours, fuel and truck maintenance).

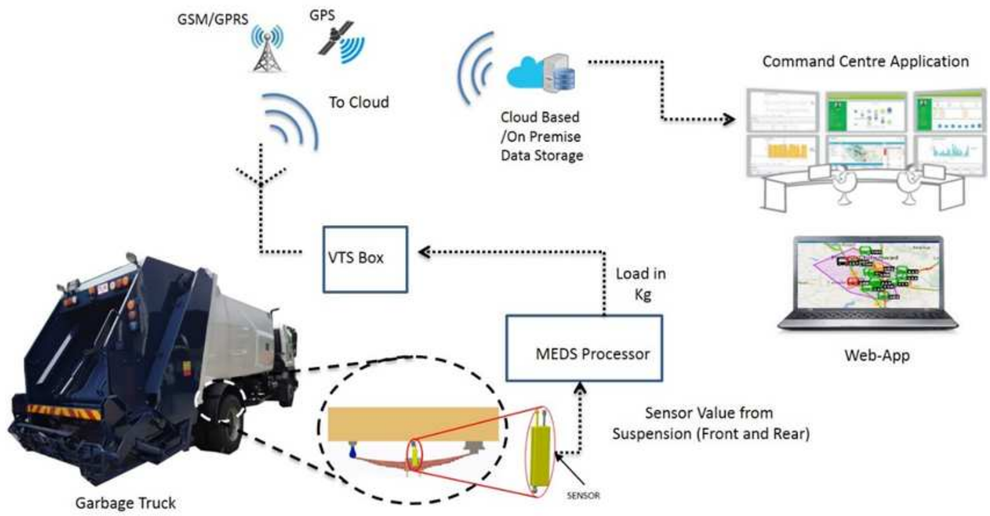


**Figure 5. Garbage truck suspension monitoring**

- Significant reduction in overflowing rubbish bins: overflowing rubbish is a breeding ground for bacteria, insects and pests because of accumulated rubbish. Its a public nuisance and unpleasant for residents and visitors to our garden city.

- Identifies the use (or misuse) of public rubbish bins: for example, sudden spikes in rubbish levels at night can indicate that household rubbish is being dumped illegally from residents or campers.

CCC selected local company PiP IoT for a pilot to test new sensor technology in the combat against overflowing rubbish bins with the objective of creating operational efficiencies and an enhanced streetscape for citizens

PiP IoT has rolled out LevelSense™devices to ten bins around rubbish trouble spots, including a city park and a retail area. Notifications are sent to contractors phones and bin status across the city can be viewed in an online dashboard telling them how full each bin is (see figure 4, right). Rubbish levels are also being tracked, providing a graphic illustration of when the bins are used most.

**2.3.3. Using ACO Update Rules** For finding the optimal route connecting the full (or almost full) containers, Christchurch makes use of ACO as well. Th algorithm makes use if two types of locally accessible information:

1. problem-specific information, i.e., distance among cities, and
2. information added by ants during previous iterations of the algorithm, such as pheromone values.

While building a solution, each ant collects information on the problem characteristics and on its own performance, and uses this information to modify the representation of the problem. The modified representation is the new environment shared by all the ants and seen locally by other ants. This form of indirect communication mediated by the environment is called stigmergy, and is typical of social insects [**?**].

The waste collection problem in Christchurch poses a dynamic instance of the TSP, where cities are represented by containers and containers which are (almost) empty are not collected. Containers which are full or almost full will appear in the bin collection route dashboard and will be collected. ACO has shown good performance on randomly generated problem instances and on real-world instances as well. This makes it the perfect algorithm for Christchurch's dynamic vehicle routing scenario.

## 3. The greedy approach to the travelling salesman problem

**Steps**

1. Calculate transport times and distances between all stops, e.g. pickup and destination locations, as well as vehicle start locations.

2. Based on the information collected in step 1, generate routes and assign vehicles. The primary

constraint is that the total travel time for the first person to board the vehicle should not exceed 1.5 times the time taken for him to reach the destination had he chose his direct transport option.

3. To go through possible options we have used simulated annealing and threshold-accepting algorithms. For each node (stop), the eligible (satisfying time & capacity constraints) neighbors are evaluated to form paths and the ones which meet the criteria (serve maximum demand with shortest path) are selected. The acceptance criteria considers aspects like remaining capacity, direction deviation from the destination and total travel time.

4. Since this is a greedy approach, this algorithm is susceptible to choice of first location and hence a customized logic is developed to run it for a set of feasible origins and best possible solution which gives minimum number of resources is chosen.

### 3.1. Application 1: The Malaysia Garbage Initiative

Malaysia is one of the trend-setters concerning smart city activities and their garbage collection initiative is unique in multiple aspects. The city does not only encompass the collection of garbage in a smart manner but it also takes into account the fleet of collection trucks and their maintenance to ensure availability and better framework to manage the contingencies that may occur.

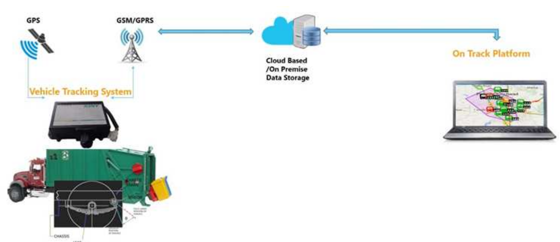


**Figure 6. Real-time routing**

For monitoring the garbage collection, Malaysia's garbage trucks are equipped with leaf spring sensors measuring the vehicles' suspension levels. The measured data and the GPS positions of the respective trucks can be visualized in a Web App and they are also used as input of the command center application for coordinating the trucks dynamically. The other parameters processed by this application comprise static information, such as the truck parking locations, the bin locations and time estimates before a bin is full. Using these data, the Malaysian authorities can observe the amount of the loaded garbage at each location, which helps them to avoid vehicle overloading and truck breakdowns. Figure 5 shows how the components of this IoT infrastructure work together. The MEDS (*Mechanical Engineering Design Service*) is was developed by `kpit.com`'s engineering team to connect a microcontroller to the VTS (*Vehicle Tracking System*). This microcontroller converts the measured deflection on the leaf suspension to payload weight and uses the GSM network to transmit the data back to the backend server.

The vehicle tracking system makes use of geo fencing to track discrepancies between the planned and the actually driven routes (see figure 6). Comprehensive reports of every vehicle's trip history, map views and the cargo weight details are available in the Web App at the click of a button.

### 3.2. Application 2: Optimized Bus Routing for Collecting Employees in Bangalore

The costs of transporting employees is a concern for many companies. Especially in India, where the number of employees is high in many of the companies these costs are huge: Employee transport costs are the third highest cost after employee salaries and rentals. Therefore, minimizing these costs is of vital importance.

The `kpit.com` initiative was undertaken to find minimal routes and also minimize the number of vehicles required to transport employees located at various location in the metro city of Bangalore.

**Inputs**

- Demand at every stop
- Stop locations
- Available fleet and their corresponding capacities

**Constraints**

- Travel time for the first passenger who boards the vehicle should not be more than 1.5 times the time he would take to reach office using his own transport.
- The total demand should not exceed the capacity of vehicle

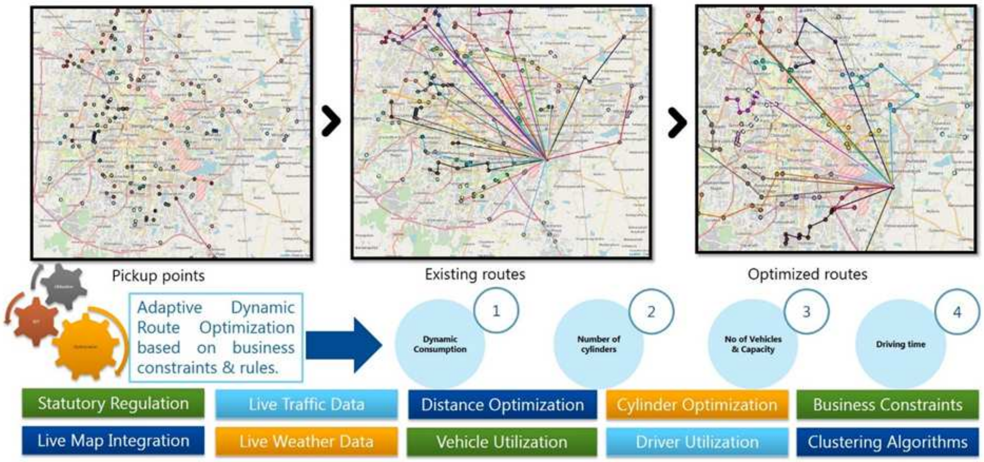


**Figure 7. Employee collection**

## 4. Conclusion and Future Perspectives

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

## References


[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Sciences*, vol. 6, no. 1, 1959.

[2] J. Chen, R. Bai, H. Dong, R. Qu, and G. Kendall, "A dynamic truck dispatching problem in marine container terminal," *Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, Dec 2016.

[3] Hyundai Motor Company, "Garabge Truck Press Pack." Online e-Brochure `http://www.hyundai.com/gd/en/find-a-car/garbage-truck/press-pack.html`, 2018.

[4] V. V. Burkhovetskiy and B. Y. Steinberg, "Parallelizing an exact algorithm for the traveling salesman problem," *Procedia Computational Science*, vol. 119, no. C, 2017.

[5] Volvo Group, "Volvo pioneers autonomous, self-driving refuse truck in the urban environment." *Press release* `http://www.volvogroup.com/en-en/news/2017/may/news-2561936.html`, 2017.

[6] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," 1997.

[7] N. V. Karadimas, N. Doukas, M. Kolokathi, and G. Defteraiou, "Routing optimization heuristics algorithms for urban solid waste transportation management," *Transactions on Computing*, vol. 7, no. 12, pp. 2022–2031, 2008.

[8] D. Burger, J. Weiß, A. Sarkar, K. Kirsch, and J. Dünnweber, "Combining fill-level sensing with route optimization for a more efficient waste collection," *18th European Conference on Digital Government (submitted)*, 2018.

[9] A. C. Lundin, A. G. Özkil, and J. Schuldt-Jensen, "Smart cities: A case study in waste monitoring and management," in *50th Hawaii International Conference on System Sciences*, 2017.

[10] Z. Ismail and S. L. Loh, "Ant colony optimization for solving solid waste collection scheduling problems," *Journal of Mathematics and Statistics*, vol. 5, pp. 199–205, 03 2009.

[11] S. Sharmin and S. T. Al-Amin, "A cloud-based dynamic waste management system for smart cities," in *Proceedings of the 7th Annual Symposium on Computing for Development*, pp. 20:1–20:4, ACM, 2016.

[12] J. L. Carlson, *Redis in Action*. Greenwich, CT, USA: Manning Publications Co., 2013.

[13] G. Reinelt, "TSPLIB - A t.s.p. library," Tech. Rep. 250, Universität Augsburg, Institut für Mathematik, Augsburg, 1990.

[14] D. Applegate, "Concorde - a code for solving traveling salesman problems." `http://www.math.princeton.edu/tsp/concorde.html`.

[15] D. Angus and T. Hendtlass, "Dynamic ant colony optimisation," *Applied Intelligence*, vol. 23, pp. 33–38, Jul 2005.