IIT PALAKKAD

Project Report
EE2170 - Mini-Project

Smart Cities: Garbage Management System

Ahmed Zaheer Dadarkar[02] , Devansh Singh Rathore[11], Priyanshu Shrivastav[22], Rakesh Kumar[24], Shruti Umat[27], and Vaibhav Jindal[29]

Batch of 2017, Department of Computer Science and Engineering,
Indian Institute of Technology Palakkad, India

October 19, 2018 @ Electronics Lab, IIT Palakkad

Problem statement and abstract of the project

Swachh Bharat Mission (SBM) is a national campaign by the Government of India, covering 4041 statutory cites and towns. Main goal of this mission is to promote cleanliness in cites, to have clean streets, roads, and infrastructures. For this purpose an efficient and effectve garbage collecton, transporting, processing, and disposal infrastructure is essential. This mini-project is aimed to design a central controller (digital system) and associated sensor network for such a garbage management system that can be deployed in smart cites.

1 Introduction

The dustbins, referred to as "collection points", are assumed to be distributed throughout the city with waste being segregated in the form of biodegradable wastes, non-biodegradable wastes and metallic / e-wastes along with an arsenal of trucks with varying capacities to collect garbage. With the collections points to be covered spanning miles in the city, accumulating garbage in an unplanned manner can take a toll over the system, increasing operational cost and collection time immensely.
The task is to make the process of garbage collection centrally controlled, automated and efficient, minimizing the time of collection by prioritizing locations and reducing significantly the distances covered by all trucks as a whole.
Since digital systems are low cost, versatile, robust in functioning and comparatively easier to design, the above objectives can be achieved by deploying a digital system which implements a basic algorithm on limited number of collection points. The collection points can always be increased with an increase in hardware and minor expansion in code yet the underlying logic would remain the same.

2 Design Objective

The following design objectives are kept in mind while designing the central controller:

i. We assume that there are only three types of bins (waste), distinguished by numbers `1, 2 and 3.`
ii. Each bin has an urgency value associated with it (boolean value).

iii. The dustbins output one among the three states: empty, half full/empty, completely full.

iv. We assume that there are trucks of three different capacities: One which can take a maximum of three dustbins, second which can take a maximum of two dustbins, and the last which can take a maximum of one dustbin completely.

v. To build a central controller which takes the urgency of each can, coordinates of each truck and can, the types of trucks available along with their sizes and maps each of the cans to a truck for collection.

vi. The following conditions must be satisfied: -

1. The design should have minimum 30 number of collection points.
2. The design must be scalable to larger numbers of collection points and trucks.
3. The collection must be prioritized based on given factors like sensor outputs and collection time.

vii. The design should be as efficient as possible.

3 Input and Processing

I. Input from each dustbin: -

II. (**coords, t_1u, t_2u, t_3u, t_1s, t_2s, t_3s)**

(a) **coords**: Coordinates of each bin

(b) **t_iu**: Urgency of ith type of dustbin

(c) **t_is**: State of ith type of dustbin

III. Input from each truck:

(a) **coords**: coordinates of each truck

(b) **type**: Type of the truck

(c) **cap**: capacity of the truck

IV. The first priority is based on the urgency of clearing the can (type specified) which is a binary value. This would be used to find the highest priority bin(s).

V. The input given to the central controller consists of coordinates of each truck and can with respect to the center. This input would be used to compute pairwise distances of each can from each truck. This would set the second priority.

VI. The type and size of garbage truck required by each can sets the third priority.

4 Main Algorithm

After grouping the bins with the same urgency, one among the highest priority bins is chosen. We wish to find the best suitable truck needed to clear it up.

Here, best suitable truck means:

- The truck which can carry maximum load it can carry in its path including the prioritized dustbin and,
- If there exists several trucks satisfying the above criterion, then the nearest truck will be the best suitable one.

Each bin can have 3 states: Empty, half-filled, completely filled.

According to the size of the each truck, number of possible states (in terms of load, unordered) it can have is:

A low capacity truck (can carry 1 completely filled bin) can have 3 states.
`states = { i*(max/2) : i ∈ [0,2] and i ∈ Z}`
`max` = maximum capacity of a dustbin

A medium capacity truck (can carry 2 completely filled bins) can have 5 states.
`states = { i*(max/2) : i ∈ [0,4] and i ∈ Z}`
`max` = maximum capacity of a dustbin

A high capacity truck (can carry 3 completely filled bins) can have 7 states.
`states = { i*(max/2) : i ∈ [0,6] and i ∈ Z}`
`max` = maximum capacity of a dustbin

e.g., if `max` = `20 kg`, then
`LOW` capacity truck can hold `0, 10, 20 kg.`
`MEDIUM` capacity truck can hold `0, 10, 20, 30, 40 kg.`
`HIGH` capacity truck can hold `0, 10, 20, 30, 40, 50, 60 kg.`

We will receive the location of every point (dustbins and trucks). Here onwards, nodes will refer to those points. A truck moving to target a bin can carry loads in its way if the difference of the truck's current capacity and target dustbin load is more than the path (extra) bin i.e it will collect garbage from a node on its way only if collecting garbage from the current node doesn't occupy the space intended for the target node. If the truck fulfills the above criterion, it will pick up the load and will move towards the target. The truck capacity will be updated.

Since there are n trucks, the best truck has to be chosen, we can use Dijkstra's algorithm[1] to solve the problem.

1. All trucks will act as sources.
2. Target will be the highest priority bin(s).
3. Each truck will have states as defined above.
4. A truck reaching a dustbin can have any of the above states. Its state will change depending upon whether it picks the dustbin or not. For each states at a node, we will store the information of nearest truck with same state and whether the current dustbin is picked up or not.
5. After running the algorithm, we will choose the maximum state (maximum load state) of the target dustbin and the information about the truck and the extra dustbins to be picked. Location of the extra dustbins and the path used can be stored by some modification in the code.

1 Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph with worst case performance O(|E| + |V| log |V|) (where |V| is the number of vertices and |E| is the number of edges).
Source : https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

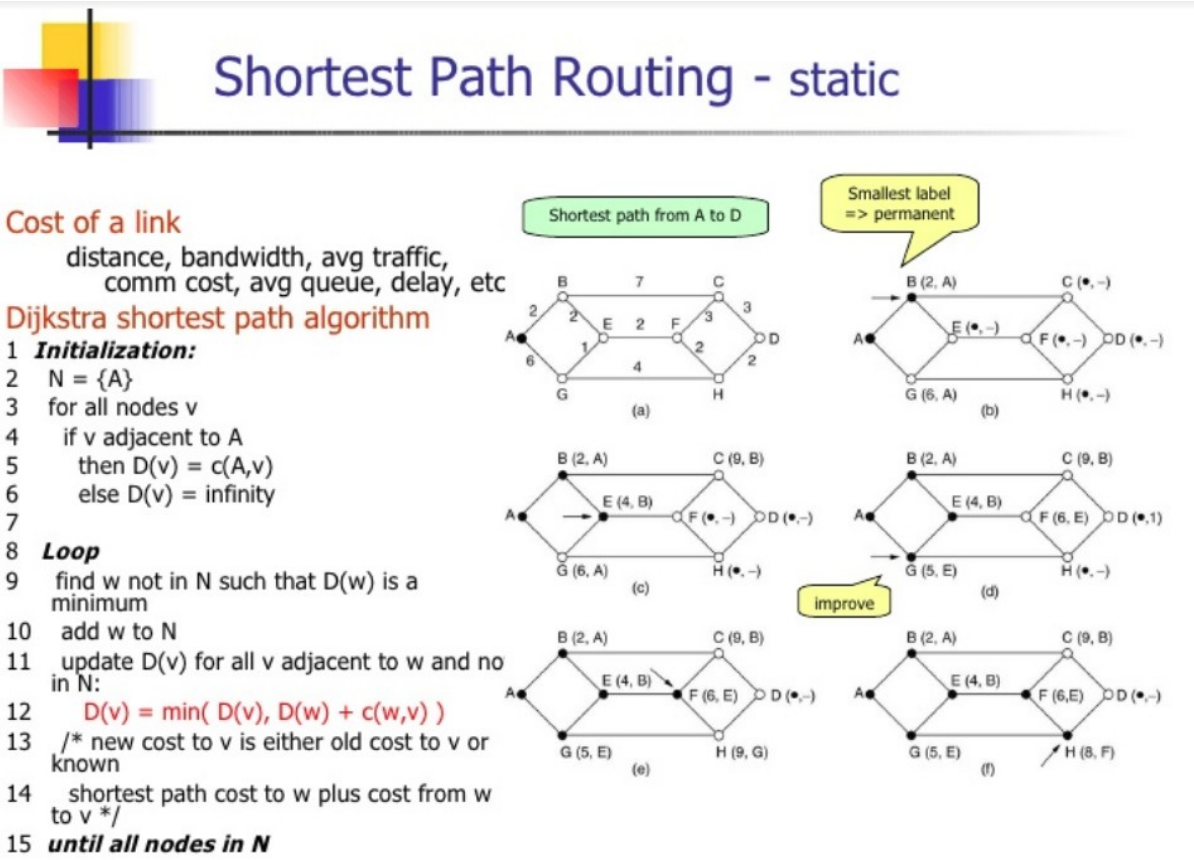

Fig 1:Pseudo Code for Dijkstra's Algorithm

5 Output and Display

1. The output from the main algorithm consists of each truck mapped to the path which it would take and the dustbins it would cover in its path.
2. Given a truck, those LEDs would light up which represent the dustbins it would be collecting the garbage from.

6 Design Justification / Discussion

1. Priorities of all dustbin outputs for a day have been taken into consideration.
2. For each bin, the most appropriate truck has been chosen by considering the truck's size and type.
3. The design is flexible to any number of nodes (dustbins) as Dijkstra's algorithm can be used for any number of nodes to calculate shortest distances of nodes from trucks.
4. Dijkstra's algorithm ensures mapping of trash can to closest truck in its path in the graph provided.

7 Loopholes/Drawbacks:

As priority has been defined for each day (which remains fixed for that day), there can be multiple dustbins which become full on the current day itself and must be given higher priority than the present dustbins but this will not work in accordance to our algorithm.

Eg. Consider the case that for the present day, the dustbin with the highest priority is partially full. However, some of the dustbins become full on the present day itself but those dustbins will be processed the next day according to our design.

If the type of truck differs from the one required by a dustbin, a required truck with the same type must get empty for the garbage to be collected.

Eg. Consider one specific type of waste being in high quantity, then only one type of truck will be on work and rest of the trucks, despite being empty, will not work. This may cause overload on some specific kind of trucks.

Fig 2: Dustbin (sources : Google Images)

8 Implementation details

i. The implementation of the digital system requires use of a `Zybo board (FPGA)` for the central control unit.
ii. Each node (trash can) in the city would send the tuple: (**`coords, t`$_1$`u, t`$_2$`u, t`$_3$`u, t`$_1$`s, t`$_2$`s, t`$_3$`s`**) to the central controller.
iii. **`Ultrasonic sensors`** can be installed in the inside part of each dustbin lid to detect the garbage level.
iv. The central controller would process this input and find the pairwise distances, then call the main algorithm with the given inputs.

v. The output from the main algorithm consists of each truck mapped to the cans that it would cover.

vi. A set of LEDs for each truck wil be used to represent the garbage nodes. Those LEDs would glow which would be covered by the truck.

Fig 3: Zybo Board

9 Conclusion

- Using the method specified above, we design an efficient garbage collection system with proper transportation.
- The algorithm we used (Dijkstra's Algorithm) for checking the priority of each can is based on important elements of graphs theory, based on which Google maps algorithm is designed.
- It is not only efficient in time but also in the amount of human resource requirement (less humans are required to work on).
- It is an on board surveillance system based on the input data we get for the cans for garbage collection.
- We can add a new can (node to the graph) at any point of time which makes it more easy to use, and scalable to any number.
- Using different trucks for different types of garbage collection which helps not only in categorising garbage and its management but also makes it easier for its processing after the collection.
- Collected garbage can be efficiently used as important `3Rs` i.e. `Reduce, Reuse and Recycle.`

NOTE: Different kinds of wastes like Biomedical Waste generated from the hospitals and factories should be treated without polluting the environment. Safely disposal of waste must be done according to the CPCB guidelines.

10 Acknowledgements

Prof. Sabu Emmanuel, Ms. Anuja Menokey, Mr. Renjith S
Indian Institute of Technology, Palakkad