PRODUCT DELIVERY OPTIMIZATION

Prof Ratna Nayak[1], Vedank Vekhande[2], Bhavya Sheth[2], Rohan Dhumal[2], Prashant Patra[2]

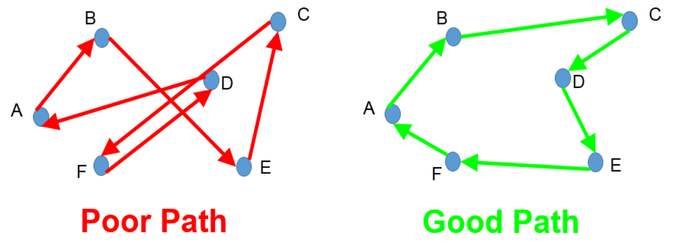
[1]Assistant Professor, Computer Department, KJSIEIT Sion, Mumbai, Maharashtra, India.
[2]Student, Computer Department KJSIEIT Sion, Mumbai, Maharashtra, India.

Abstract - *In Today's Modern World of Digital Technology we often come across the issue of Transportation especially while travelling on multiple destinations. Transportation can be related with delivery of a product from E-commerce sites, Public Transportation like School bus, Garbage collector vans, etc. So, it pretends to be quite tedious task of planning the entire route, especially when there are much greater number of locations. So, in our approach we used Dijkstra's algorithm to calculate distance between the two distinct pairs of given input locations. Recursively calculating the distance between every two pairs and selecting the one with least distance is main objective of my approach. This would be followed until all the given input locations would be visited by the user. The process of distance calculation is automated using the RPA software calculation, elsewise Customer would have needed to manually enter the locations on web and analyse it for endless hours. Our designed Application will just locations as input and display the optimized sequence of locations to the Customer. This will save not only Time but Money and Fuel too. The application was much more needed for since it will be saving the valuable resources of environment.*

Key Words: **RPA, Orchestrator: - Software Bots API:- Application Program Interface, GA:- Genetic Algorithm , CRM**

1. INTRODUCTION

This document is template. We ask that authors follow some Product Delivery Optimizer is a simple API like web application which would be assisting anybody either personal or commercial agent desires to travel on multi-route locations in least possible time or saving time or else both. The application would be helping such desirous people to by planning their journey in real time, without really bothering about complex planning.



A) Robotic Process Automation (RPA)

Robotic Process Automation (RPA) is a latest software tool in the market for dealing with the repetitive tasks and processes mostly associated with the computer software. In Today's world of Digitization tons of data is generated in the form of Application Forms, Queries, Excel sheets, etc. A lot of Manual Workforce is required to fill up these forms or to perform the same repetitive task. RPA comes into the picture for such a set of repetitive tasks. Whether be it filling online forms, excel sheets, emails, continuous monitoring, web trafficking or any such work.

RPA recommends items by matching the attributes of a product with the user's profile.

Advantage

- Makes Repetitive tasks easier of user preferences.
- Structure is simple to visualize.
- Wide support of Applications and API.

B) Designing Elements of RPA

i) Flowcharts

Flowchart is one of the designing platforms where typically like an algorithm our working model flow. Using Flowchart structure in RPA helps us in visualizing the flow of model easily. Mostly Flowchart is used to in the case when working with the loops like structure or repetitive subroutine calls, etc.

Advantages

- High performance.
- Clean Visualization.

ii) Sequences.

Sequences unlike the flowcharts helps to embed the multiple processes one inside the other. Unlike Flowcharts they don't allow full view of all the processes at the same time but possess much higher speeds of execution. Also, they can't support looping like jumping back to the passed location.

Advantage

- Allows multiple task embedding.
- Eliminates Loops.
- Accurate.

Disadvantages

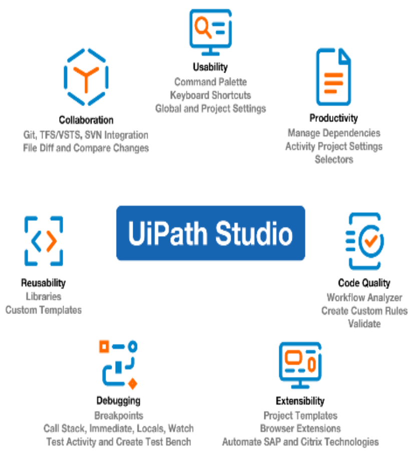
- Jumping back not allowed.
- Scalability problem.
- Multiple variables are required.

C) State Machines

A state machine is a type of automation that uses a finite number of states in its execution. It can go into a state when it is triggered by an activity, and it exits that state when another activity is triggered.

Another important aspect of state machine are transitions, as they also enable you to add conditions based on which to jump from one state to another. These are represented by arrows or branches between states.

There are two activities that are specific to state machines, namely State and Final State, found under Workflow > State Machine.



Uipath (RPA) Robots:

The Robot is UiPath's execution agent that enables you to run processes developed in Studio. Robots need to be connected to Orchestrator in order to execute processes or they have to be licensed locally (read more about licensing). The type of Robot you have at your disposal is determined by the license, while the Robot service is determined by the deployment.

The Robot is split into four components, each being dedicated to a particular task in your automations. The Robot components are as follows:

- Service
- Service Mode
- User Mode
- Executor
- Tray
- Command Line Interface

Having the Robot components split as explained above helps developers, support users, and computers easily run, identify, and track what each component is executing. Special behaviours can be configured per component this way, such as setting up different Firewall rules for the Executor and the Service.

UiPath Orchestrator is a web application that enables you to orchestrate your UiPath Robots in executing repetitive business processes.

Uipath Orchestrator:

Orchestrator lets you manage the creation, monitoring, and deployment of resources in your environment, acting in the same as an integration point with third-party solutions and applications.

UiPath Orchestrator Use Cases

UiPath's Orchestrator power comes from its capability of managing your entire Robot fleet. Attended, Unattended, Nonproduction, Studio, or StudioX Robots - they can all be managed from this centralized point.

Attended - This type of Robot is triggered by user events, and operates alongside a human, on the same workstation. Attended Robots are used with Orchestrator for a centralized process deployment and logging medium.

Unattended - Robots run unattended in virtual environments and can automate any number of processes. On top of the Attended Robot capabilities, the Orchestrator is responsible for remote execution, monitoring, scheduling and providing support for work queues.

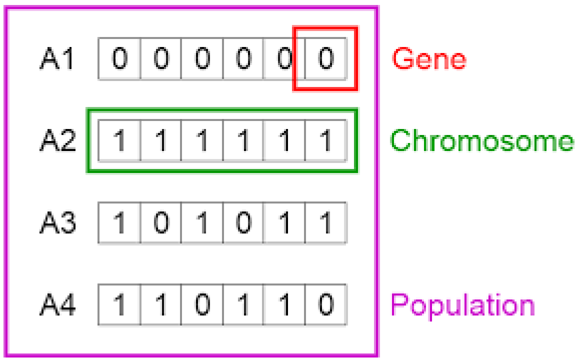
Studio / StudioX - has the features of an Unattended Robot, but it should be used only to connect your Studio or StudioX to Orchestrator for development purposes.

NonProduction - similar to Unattended Robots, but they should be used only for development and testing purposes.

You are able to run debugging in Studio with all types of Robots.

Genetic Algorithms: -

Genetic algorithms (GA) like neural networks are biologically inspired and represent a new computational model having its roots in evolutionary sciences. Usually GAs represent an optimization procedure in a binary search space, and unlike traditional hill climbers they do not evaluate and improve a single solution but a set of solutions or hypotheses, a so-called population.



The GAs produce successor hypotheses by mutation and recombination of the best currently known hypotheses. Thus, at each iteration a part of the current population is replaced by offspring of the most fit hypotheses. In other words, a space of candidate hypotheses is searched in order to identify the best hypothesis, which is defined as the optimization of a given numerical measure, the so-called hypothesis fitness. Consider the case of function approximation based on given input-output samples: The fitness is the accuracy of the hypothesis (solution) over the training set.

The strength of this parallel process is enhanced by the mechanics of population modification, making GAs adequate candidates even for NP-hard problems. Mathematically, they are function optimizers and they encode a potential solution based on chromosome-like data structures. The critical information is preserved by applying recombination operators to these structures. Their most interesting properties are [2]:

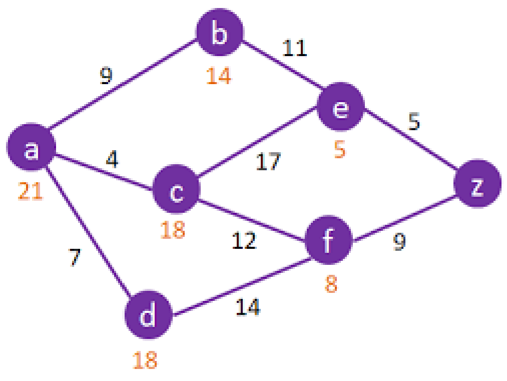
•Efficiency.

•Simple programmability.

•Extraordinary robustness regarding the input data.

The most important property is robustness, and this represents an emulation of nature's adaptation algorithm of choice. Mathematically, it means that it is possible to find a solution even if the input data do not facilitate finding such a solution.

A* Algorithm:



Google Maps uses A* algorithm for finding the shortest path and alternates routes in real time. A* algorithm is an advanced form of Breadth first search. It avoids costly path and choose the most promising path. It is a very smart algorithm. It is used approximate the shortest path in real-life situations, like- in maps, games where there can be many hindrances. It is formulated in terms of weighted graphs in case of google map this weight is travel time. Starting from a specific node (source node) of a graph, it constructs a tree of paths starting from that node, expanding paths one step at a time, until one of its paths ends at the predetermined destination node.

At each iteration of its main loop, A* needs to determine which of its partial paths to expand into one or more longer paths. It does so based on an estimate of the cost (total time taken) still to go to the goal node. Specifically, A* selects the path that minimize. f(n)=g(n)+h(n)

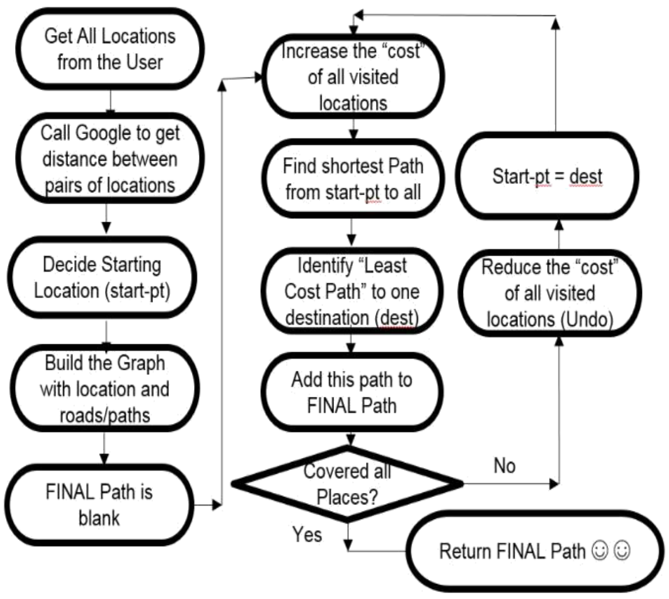
where n is the destination node on the path, g(n) is the cost of the path from the start node to n, and h(n) is a heuristic that estimates the shortest path from source to the destination. The heuristic is problem-specific. In this case it is time taken to reach somewhere.

Why A* is better than other: -

A* is the most popular choice for pathfinding, because it's fairly flexible and can be used in a wide range of contexts. A* is like Dijkstra's Algorithm in that it can be used to find a shortest path. A* is like Greedy Best-First-Search in that it can use a heuristic to guide itself.

2. DESIGN

The Designing consists of a looping algorithm of whole process as shown in the following flowchart presentation


Get All Locations from the User
Call Google to get distance between pairs of locations
Decide Starting Location (start-pt)
Build the Graph with location and roads/paths
FINAL Path is blank
Increase the "cost" of all visited locations
Find shortest Path from start-pt to all
Identify "Least Cost Path" to one destination (dest)
Add this path to FINAL Path
Covered all Places?
Yes
No
Start-pt = dest
Reduce the "cost" of all visited locations (Undo)
Return FINAL Path ☺☺

I. EXISTING SYSTEM

Many other Applications software and API are available on the Internet for Managing and Planning the Routes commercially. Example includes the Routific and Locus.sh are the popular API's available in the market. Locus.sh manly focuses on the Customer Relationship Management, along with route optimization, but its main focus is for generating optimized path for the vehicular transportation. They both use the Deep learning algorithms and techniques by learning form the users navigating data.

Google Maps is great when you have a small number of deliveries to make. It's free, fast, and extremely user-friendly. To use Google Maps as a route planner, look no further than this helpful guide Google has put together.

With that said, there are some limitations when you're trying to plan routes for deliveries.

1.Your routes need to be 10 stops or less

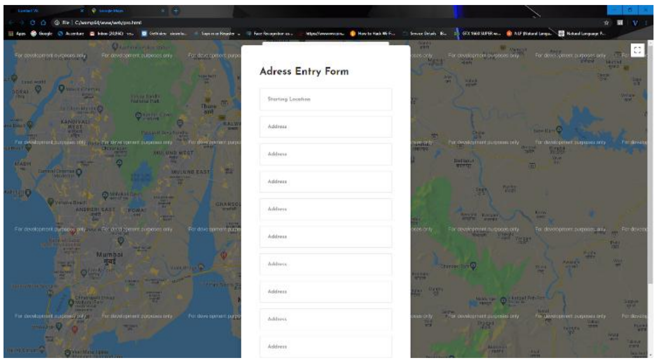
2.You can only plan for 1 driver at a time

3.You can't optimize routes using constraints like delivery time windows, vehicle load capacities, driver breaks, etc.

4.You need to eyeball and manually determine an efficient order for your stops

II. PROPOSED SYSTEM

Our project initiates with taking the inputs from simple easy to use online web-application form grabbing the data from the user. The data is in the form of the addresses of the locations which could be submitted by the user. Once the form is submitted, the data would appear in the database of server machine where the RPA software and related databases would be residing. As soon as the user clicks on the continued monitored "submit" button of the RPA main RPA workflow is triggered by the server ".bat" file.



Adress Entry Form

Starting Location

Address

Address

Address

Address

Address

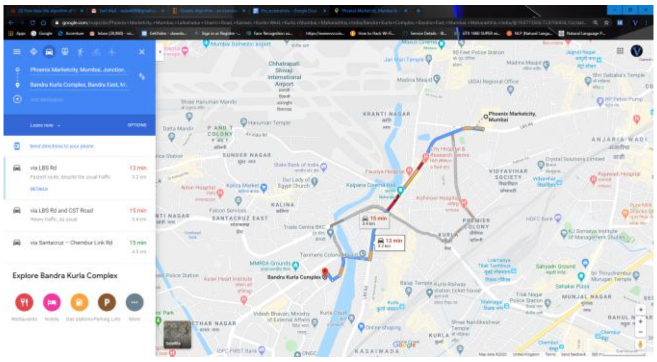
Address

Address

Address

Address

The RPA workflow rapidly starts automatically calculating the distances between the pairs of locations. Each pair's location is stored, sorted and a sequence according the smallest distance is generated. This new generated path is being stored in parallel in the database. The process is in beta version and supports upto the 20 locations at the time. The distance calculation is being implemented on the available Google Maps API. The sorted sequence is conveyed to the user either on the provided mobile number or the email-id depending upon the regional availability.



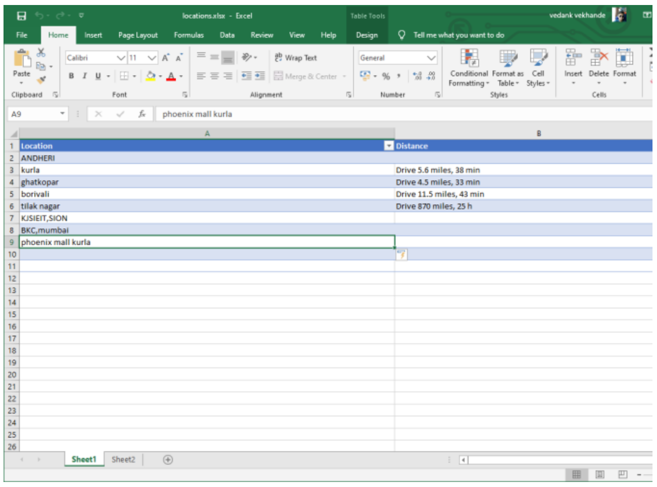
The proposed system contains the following important parameters:

1. Varying Distance modes:

- Short distance travel: We chose different addresses in a specific city like the Mumbai City. Generally, we used the addresses of local fire stations and police stations in the city.
- Medium distance travel: We chose different addresses in different cities in a specific state. For example, in some runs, we chose following 7 cities in the state of Mumbai Suburban like Thane, Navi Mumbai, Palghar, etc.
- Long distance travel: We chose different destinations across America, that spans through multiple states. For example, in some runs, we chose the city centers of following cities: Delhi, Chennai, Kolkata, Goa, etc.

***B.* Varying Number of Locations**

1. The data which is the location/address of the user in Realtime.
2. A* algorithm is used as a trial algorithm in the initial stages for the performance analysis of the designed application.
3. Based upon the accuracy of the A* algorithm and efficiency, further modification in the algorithm will be performed.



locations.xlsx - Excel

A9 phoenix mall kurla

	A	B
1	Location	Distance
2	ANDHERI	
3	kurla	Drive 5.6 miles, 38 min
4	ghatkopar	Drive 4.5 miles, 33 min
5	borivali	Drive 11.5 miles, 43 min
6	tilak nagar	Drive 870 miles, 25 h
7	KJSIEIT,SION	
8	BKC,mumbai	
9	phoenix mall kurla	

3. CONCLUSION

IRJET Our primary goal is to provide this tool to different companies or organizations that have people who need to visit multiple destinations. Using this tool, they can determine the efficient path quickly, and that eliminates the need for them to manually identify a path. As a result, this approach allows people and organizations to save time, and money which could be used in other places. This proposed methodology is practical and extremely easy to use. Our experimental data-set indicates that our proposed approach determines a path within a very short amount of time, and the path is almost always the best possible path that a human could have generated by using trial and error approach over several hours and days of time.

REFERENCES

[1] E. W. Dijkstra. A note on two problems in connexon with graphs. Numerische Mathematik, pages 269–271, 1959.

[2] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107, July 1968

[3] P. Siy. Road map production system for intelligent mobile robot. In Proceedings. 1984 IEEE International Conference on Robotics and Automation, volume 1, pages 562–570, Mar 1984.

[4] M. Noto and H. Sato. A method for the shortest path search by extended Dijkstra algorithm. In Systems, Man, and Cybernetics, 2000 IEEE International Conference on, volume 3, pages 2316–2320 vol.3, 2000.