

A BI-CRITERIA SCHEDULING FRAMEWORK FOR THE SUPPLY CHAIN MANAGEMENT OF MOBILE PROVIDERS

Göktürk Gezer, İlker Yaz, Hasan Mert Taymaz, Tansel Özyer
Department of Computer Science, TOBB ETÜ, Ankara, Turkey

Reda Alhajj

Department of Computer Science, University of Calgary,; and Dept. of Computer Science, Global University, Lebanon

Keywords: Scheduling, delivery, agent, PDA.

Abstract: In delivery sector answering the requests of customers quickly and efficiently became very important for recent years. Thus people who are working in delivery sector began challenging to build well settled synchronization for more complex delivering system. In this synchronization they try to keep customers information, providers (who answering requests) information, divide requests for providers and inform them about their recent jobs, provide fast and efficient delivering system. All of these lead to acquire customer pleasure with firm pleasure. We believe that using mobile device such as PDA with a host PC for building well settled synchronization with providers and host is simple but useful way. The aim of host is arranging received requests and sending them with their information to convenient providers by considering all providers position and their job intensity. So request would be answered as quickly as possible. Also providers can access information about their jobs and the shortest path for reaching these jobs by using their PDA. Our system makes all things simple and easy for people. Providers don't have to care about how to reach their jobs and workers arrange requests don't have to consider about convenient providers.

1 INTRODUCTION

Mobile devices are employed in several real life applications. Supply chain management is one of them. It includes the movement of production from supplier to customer; submitting the delivery with delivery status; and financial flow such as terms and payments scheduled. Vehicle routing problem paid an attention by researchers (Bodin et al., 1983) (Desrosiers et al., 1995). Vehicle routing is important because a system needs an optimized scheduling of deliveries. It can be a service, delivery, transportation, freight. Several optimization methods have been optimized. There are some heuristic methods (Lee et al., 2001) (Savelsbergh, 1985) (Desrochers et al., 1992), Genetic algorithm based solutions (Jr. and Wainwright, 1991). It was also investigated adjunct to data structures such as trees (Muslea, 1996).

The rapid development of supply chain management between logistics and supply information chain management between providers and host, the most important problems are arranging

system for a lot of providers and customers, supply data transfer between providers-host and finding the most convenient providers for customers. In solution of these problems we use mobile devices (for providers) and PC(for host) for providing connection and through this connection after taking requests, convenient providers are selected and informed about these requests. Process of deciding the convenient provider two priorities of source management that can decide by users of system are considered. One of these priorities is choosing closest provider (by looking his last jobs location because after he answered his all jobs he will be at this location) irrespective of how many jobs he has and when he can answered the job. Second of them is choosing the provider that can answer request more quickly than others irrespective of distance between provider and job. Also we use providers' last job location instead of providers' real time location. So we don't have to connect all providers to learn their location while deciding convenient provider. When host decided convenient provider, it inform him about job and ask him if he accepts job

or not. Unless taking negative answer from provider, host adds information of the job in the provider's jobs information. If provider refuse the job, information about job send second convenient provider and ask him to accept job. So we avoid unnecessary data transfer (for example informs all providers about job) by choosing the most convenient provider for a job. Also we give a chance to users to choose their priorities (the closest or the fastest) while deciding convenient provider. These priorities make the choice convenient provider part flexible.

Information that is sent to providers includes the shortest path between him and job. Provider can see this path by the agency of map in his PDA. Also he can know estimated delivery time for jobs. In addition to this, we don't supply connection provider to provider for keeping data transfer fast between providers and host.

2 SYSTEM ARCHITECTURE

Our system does the part for preparing any map for simulation. Paths can be bidirectional besides one way or blocked at a degree. Also the implementation runs on two different types of party. Host and PDAs.

2.1 Preliminary Work

2.1.1 Road Segmentation

Distances between road pieces must be stored to compute distances between points for scheduling deliveries on the map. In order to make it possible, maps must be digitalized into memory. We have created our own data structure called "Multi Linked List" that is used to keep map locations. Figure 1 shows how the map is divided into segments conceptually.

In our application we prepared supporting software for preparing a map. It helps us code road pieces' information. By using this program we have created the necessary codes to make any map, and then our application pre-processes and stores map data in multi-linked list information. We have created the necessary codes to implement the map.

In our system, the central host and PDAs use a common map. It is a real map that takes all the connections and directions into account with road status that is caused by several reasons such as weather, pavement, traffic and etc. Before the application runs, the map should be processed to

capture the road information like a real map. Our implementation partitions the roads into segments.



Figure 1: Constructing map.

Multi linked list structure is employed to represent connections.

The suggested multi linked list data structure is a union of multi linked nodes. Each multi linked node contains following information about road piece: Road's piece number, road's length, road's activity, road's traffic intensity, list of connected road pieces to road.

2.1.2 Containers

We have created three containers to keep the data in our project. All three are explained below.

Type I Container: This container is created for encapsulating the information about job. It has types for keeping typical information and location information about the job.

Three classes are defined. They store the job information (Descriptive Information), Road Information, and Schedule Definition.

Type II Container: It contains the tree structure for our path finding algorithm.

Type III Container: It stores our multi linked list implementation for digitalizing the map. It has two classes for map and road segments each.

2.2 The Implementation

There are two main parts interacting for scheduling jobs. They are namely Host module and client module. Master-slave multi-agent approach is adopted. Host component makes computations and make decision. Although clients have the freedom of act in an environment, and clients' responsibility is to do orders received by host. Clients may accept or deny requests from the host.

2.1.2 Host Agent

Host agent is the coordinator agent for the scheduling. It undertakes the responsibilities as following:

- Editing the properties about roads. That properties include roads traffic intensity and whether it is open or not,
- Monitoring each connected provider's location in the map,
- Receiving the information about the job.
- Calculating this job's needs according to company's source management priorities,
- Finding the ways that include shortest path from each provider to customer, and then sending them with the information about job to the selected provider,
- Assigning the job to selected provider,
- Monitoring information about jobs by categorizing them into pending, completed and unassigned state,

When job has taken by provider, this job is added to pending jobs.

When provider confirm that job has finished in his program. Job is deleted from pending jobs list and added to Completed Jobs list. You can also get detailed view of the job by clicking this in the list.

2.1.3 Provider Agent

Provider agent is the client responsible for doing the job received from host. There is one host and multiple providers. Providers can get connected to the system. There is no limitation of number of providers. Client agent's functions and responsibilities are followings:

- Accepting the job received from host or denying it.
- Keeping list of pending jobs.
- Showing the information about handling job.
- Informing the host about completed jobs.

When host send a job to client a MessageBox appears on the client screen. In this screen provider

is informed about when he can start the job and when it is responsible to finish it. If provider selects "YES", job is assigned to provider and it is added to client program's pending jobs list. If provider selects "NO", job information iterates over other providers.

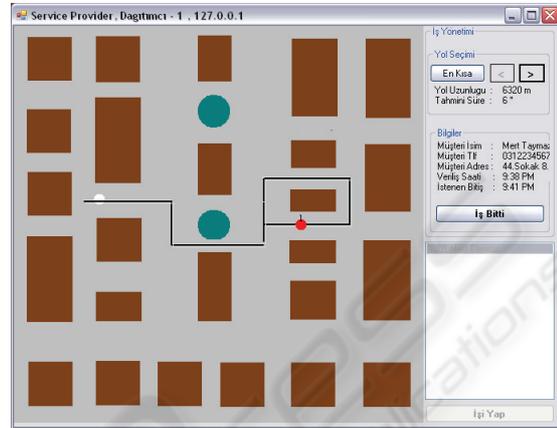


Figure 2: Proposing Road to Provider

Road manipulation is also useful property of client program (Figure 2). By clicking next, previous and "En Kısa" button, provider can iterate over roads from himself to customer (Figure 3). When provider clicks one of these three buttons, road on the map and information related to road changes according to clicked button.

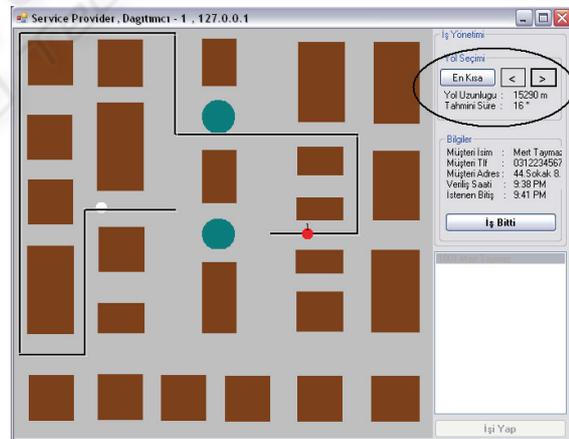


Figure 3: Listing of Optimal Paths.

When provider finishes the job he must click on the "İş Bitti" button. After clicking, information about job is removed from client program then on the host side and added to completed jobs list.

3 COMPONENTS

3.1 Host Agent Components

There are three components that are linked to host. Host is “singleton activated host” for these components. Host itself is also client for them.

ServiceProviderInfo Component: This component is responsible for keeping and serving ID and IP information about connected providers. All clients are connected to this component. They record trajectory information of clients.

JobInfoManagement Component: This component is responsible for keeping and serving job information. After host sends job information to the DecisionMaker component, decision maker tries to assign this job to client. When a client accepts the job, decision maker add this job information to the JobInfoManagement component.

DecisionMaker Component: It is responsible for making distance - time consumption calculations and assigning the job to provider by co-operating with all providers. DecisionMaker is client for JobInfoManagement, ServisProvider Info components of host and Responder component of client.

Find the Ways from Provider to Customer (*): It finds and scores all possible ways from provider to customer and sorts them in descending order.

3.2 Provider Agent Components

There are two components in client side. Client is “Singleton Activated Host” for these components.

TaskHandler: It is the main component of client program. It is responsible for keeping records of received job’s information. This information includes roads, customer information and time information.

Responder: This component is the interface between host components and TaskHandler component of client. When decision maker sends a job to client it connects to client’s Responder component, then Responder asks for approval to provider by viewing confirmation box.

4 DYNAMIC CONFIGURATION

4.1 Closing Traffic Stream on Road

In host program, operator can close particular roads for traffic stream. When road is closed, that passage won’t be used in finding algorithm. This is how we do and how it affected the algorithm below in Figure 4. You can see that closed road hasn’t used for shortest path from provider to customer though they are in the same location. This shows that our experiment result for road closing is successful.

Algorithm 1: Main Distribution Algorithm.

```
//This algorithm is responsible for creating
IsBilgileri object for each provider and sending
them to providers.
```

```
Input : TanimlayiciBilgiler object
For each connected provider
  Begin
    Retrieve providers last location
    Find the ways from provider to customer(*).
    Create YolBilgileri object from these ways
    and sort the list in ascending order
    Create IsBilgileri object and add this to
    the "Isler" list
  End;
Isler.Sort( Comparer object)
For each IsBilgileri in "Isler" list
  Begin
    Establish connection to provider's
    Responder component
    If (provider accepted the job)
      Add this job to pendings list
      Break
  End
```

In this algorithm providers last location means that, if provider has no pending jobs it last location is its real-time location, if it has some pending jobs, its last location is its last customer’s location. Comparer object that is used to sort “Isler” list is created by host according to priorities combo boxes.

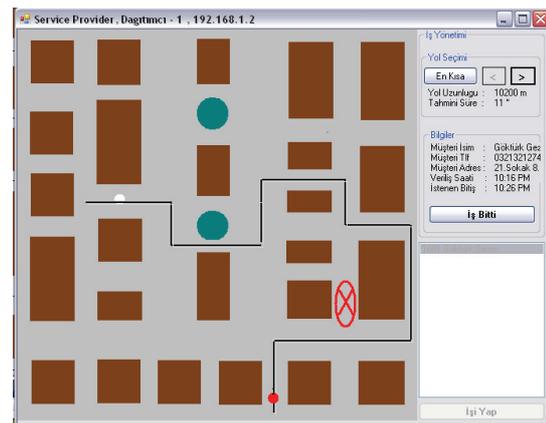


Figure 4: Path with Minimum Distance after Block.

4.2 Changing Road Intensity

Different traffic intensity of the same path changes estimated time. We have done that by changing the

route's road pieces' traffic intensity from host program. When traffic intensity for all road pieces are 50 percent, it takes 8 time units and when intensity for selected road pieces are 80 percent it takes 11 time units. This results show that our experiment for changing traffic intensity is successful.

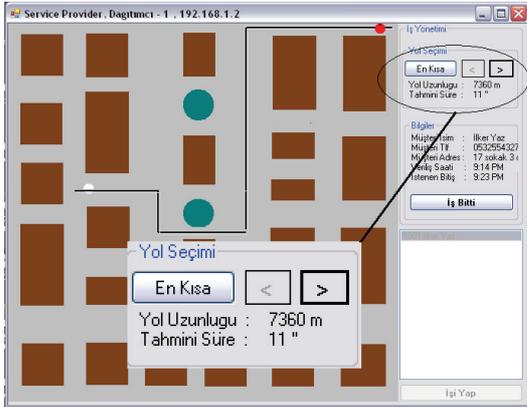


Figure 5: Route Intensity Value at 80%.

5 THE EXPERIMENTS

We have done a lot of experiments to prove our system's benefits and correctness. This application presents a framework for carrying out simulations. At the same time, this application can be controlled manually and work on real PDA devices. We have taken experiments on a map.

We assumed there are 8 and 10 providers to complete 25 jobs for the experiments. Figure 5 and Figure 7 give the meters taken per job; Figure 6 and Figure 8 give the results for the delay per job for 6, 8, and 10 providers respectively.

In each figure results belong to different cases. Random provider method is totally random. Jobs are scheduled for random providers.

Table 1 gives the results for all cases. Our method using total way taken and the delay for delivery were the main criteria for scheduling the agents for the supply chain management. In all methods increasing the number of providers increases the performance of delivery. Our algorithm is by far the most efficient system for each case. Both the delay and total meters taken (fuel consumption) are minimized. Decreasing the importance of meters taken causes total distance to increase and delay accordingly.

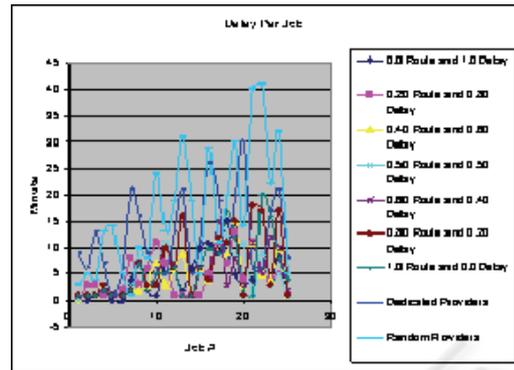


Figure 6: Service Delay for 8 providers.

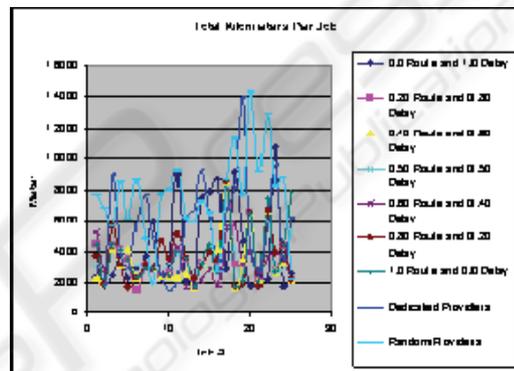


Figure 7: Road Taken for 10 providers.

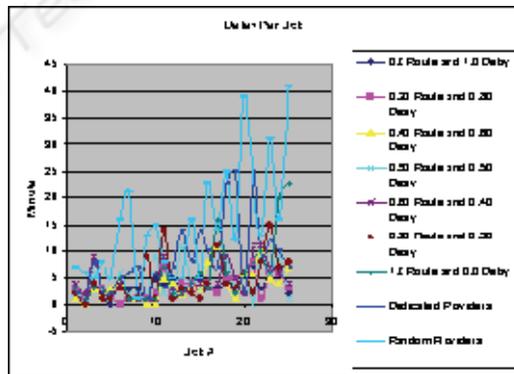


Figure 8: Service Delay for 10 providers.

Dedicated provider method shares the area equally as one provider per zone.

Others give weighted importance for two different criteria, total meters taken throughout the deliveries, and the delay for all submissions.

In this study, at the beginning we have to burn-in the system for all cases in order to obtain the maximum and minimum values to normalize the values. Delay duration is given a pre-specified unit of time, 15.

Table 1: Total Mt. and Avg. Delay For 6,8,10 Providers.

	Road Taken (meters)		Average Delay	
	8	10	8	10
0-1.0	105870	105870	4,52	3,6
.2-.8	90180	89160	4,6	3,32
.4-.6	80600	82340	4,8	3,72
.5-.5	85740	86800	4,84	4,48
.6-.4	80600	79870	4,84	4,4
.8-.2	85740	91129	6,44	4,48
1.0-0.	84880	94100	5,64	5,68
Dedicated	151431	137080	12,4	8,64
Random	195390	186430	16,72	14,88

6 CONCLUSIONS

Scheduling is an important issue and in this study, we have implemented a system that can be run for both real life system and simulation purposes. Given a map of a location, this system can be installed on PDA devices and can be used for communication between the PDA devices and the host. PDA users can communicate; accept/deny the upcoming requests for delivery, report deliveries, update traffic status, any block, and revoke any job. In addition to these, a framework for simulating the scheduling process can be configured. The number of providers, speed can be adjusted for each independently. Dynamic changes regarding the map can be set with parameters. Different jobs can be created. Finally results can be obtained with bi-criteria method. Also, random providers and dedicated provider methods have been implemented in this study. All these methods have been experimented and results convey us to use bi-criteria method for scheduling.

7 FUTURE WORK

In our project we use a small map and simulation of GPS (by clicking on map in provider system) rather than large area map and connection with GPS. In process of improving our project, large and real map can be appended the system and association between client system and GPS can be provided. Also instead of using our helper program, constitution of road piece can be done automatically by a program which can recognize road piece (may use unique colour in roads). In addition to this, our master-slave model in communication of provider and host, providers can communicate with each other and exchange work in

unexpected situation such as closed road that don't be declared before. Also the map can divide pieces and providers can be assigned his special piece where they taking request from customer in. Number of providers for each piece can be determined considering number of request for that piece. Also communication of provider can be restricted. For example only allow communication for providers that stand at the same zone. So we can avoid complex and unnecessary communication. It comes with advantage of keeping data transfer over the internet connection of PDAs fast.

ACKNOWLEDGEMENTS

The authors would like to thank to Eminay Yurtseven on behalf of Microsoft Academic Programs Management, Developer & Platform Group, Microsoft Turkiye for the support of this work.

REFERENCES

- K. Tan, L. Lee, Q. Zhu and K. Ou, "Heuristic methods for vehicle routing problem with time windows", *Artificial Intelligence in Engineering*, 15, 281-295, 2001.
- I. Muslea, "The k-Vehicle Routing Problem in Trees", M.S. thesis, Department of Statistics and Computer Science, West Virginia University, 1996.
- Savelsbergh, M. W. P., "Local search for routing problems with time windows", *Ann. of Operations Research*, 4:285-305, 1985.
- Jr., J. L. B. and Wainwright, R. L., "Multiple vehicle routing with time and capacity constraints using genetic algorithms", In *Proceedings of the Fourth International Conference on Genetic Algorithm.*, pages 452-459, 1991.
- Lawrence Bodin, Bruce Golden, Arjang Assad, and Michael Ball, *Routing and scheduling of vehicles and crews - the state of art*, *Computers & Operations Research*, vol. 10, no. 2, pp. 62 - 212, 1983.
- Martin Desrochers, Jacques Desrosiers and Marius Solomon, "A new optimization algorithm for the vehicle routing problem with time windows", *Oper. Res.*, volume 40, 2, 1992.
- J. Desrosiers, Y. Dumas, M.M. Solomon, and F. Soumis. "Time constrained routing and scheduling", *Network Routing. Handbooks in Operations Research and Management Science*, Vol. 8, pp. 35-139, North-Holland, 1995.