Towards an Intelligent Brokerage Platform: Mining Backhaul Opportunities in Telematics Data

Jonathan Muckell, Qing Cao, Patricia Mackenzie, Daniel Messier, Joseph Salvo

Computing and Decision Sciences, GE Global Research
One Research Circle, Niskayuna, NY 12309
{muckell, caoq, mackenzie, messier, salvo} @research.ge.com

ABSTRACT

In commercial transportation operations, one of the largest wasteful expenditures is the movement of tractor-trailers with little or no cargo. Analysis of inter-fleet data shows many lost opportunities for identifying backhauling loads - cargo that could have been moved by an otherwise empty trailer on its return from a delivery point to its home base. Brokerage systems that facilitate matching of load sharing and backhaul opportunities currently do not incorporate monitoring of real-time, geo-based information, analysis of historical geo-based information, and user-calibrated preferences from all brokerage participants. Future intelligent brokerage systems will need to provide a full range of services including supply chain visibility, and automated identification of potential collaborations based on historical trends. Currently, the lack of automation results in trucks pulling empty/partial cargo or deadheading (traveling without a trailer), despite potential for collaborations. In this paper, we describe an algorithm for identifying load sharing and backhaul opportunities based on the detection of patterns in large-scale, event-based telematics network data.

Keywords
Intelligent Transportation Systems, Brokerage Systems, Backhaul Analysis, Temporal and Spatial Data Mining

1. INTRODUCTION

Currently, freight brokerage systems match loads to participating partners either through individual driver's use of kiosks located at various trucks stops, or other brokerage services such as web auction systems. These rudimentary discovery mechanisms do not provide small owner-operator carriers and individual drivers a simple framework of real-time matching of loads that optimally fit into their schedules and meet personalized requirements. Larger fleets also suffer from missed backhaul opportunities due to lack of visibility into route patterns of potential collaboration partners.

In this paper, we describe an algorithm for near real-time detection of backhaul opportunities, thereby reducing wasteful transportation behavior. The next generation of intelligent transportation systems will likely include brokerage platforms that incorporate automated backhaul opportunity detection. These electronic commerce environments will bring together small businesses (carriers, drivers, shippers, receivers) to a single marketplace and offer them a comprehensive set of tools needed to facilitate context-aware, personalized, highly customizable, real time negotiations and contract agreements. Backhaul detection is a crucial feature for future brokerage systems in order to facilitate automated business partner discovery and multi-hop schedule recommendations. Larger fleets will also benefit from automated backhaul detection systems that are based on analysis of route patterns of potential partners.

Automated, near-real time detection of viable backhaul opportunities could significantly improve freight transit efficiency. Estimates suggest that 30% of all fleet trips are empty miles, the distance traveled when a truck is pulling no cargo. A 10% reduction in these trips through improved matching of potential drivers with loads would result in a net decrease of 3% of fleet miles. According to the American Trucking Association, 2.7 million class 8 trucks operate in the United States [1]. The Environmental Protection Agency estimates the average truck travels about 100,000 miles per year with a fuel efficiency of 5.5 miles per gallon [2]. Assuming a diesel fuel price of $4.75 a gallon, fuel expenses alone cost $85,000/year/truck. A 3% reduction in fleet miles would result in a net savings of approximately $2,500 for each truck or $6.75 billion for the entire US fleet. This would also reduce CO_2 emissions by 15MM tons, the equivalent of removing almost two million mid-sized cars, each traveling 15,000 miles per year, from the American roadway system. Other fuel-based emissions would also be reduced. (e.g., 9 thousand tons NOx per year). Due to the reduction in truck miles, a reduction in traffic fatalities and roadway congestion would also be expected.

To promote inter-fleet efficiency, intelligent brokerage systems will need to incorporate monitoring of real-time, geo-based information, analysis of historical geo-based information, and user-calibrated preferences from all brokerage participants. In recent years, telematics systems that integrate wireless communications with sensor-based monitoring and location-aware applications have been widely deployed for mobile asset tracking and condition monitoring. With massive volumes of disparate data being generated by such systems every day, advanced analysis of large-scale telematics network data is required to obtain the information needed for intelligent brokerage systems. The data in this paper is collected from a telematics network that monitors hundreds of thousands of mobile assets. This is one of the first instances of mining real world data from a large, event-based telematics platform.

The primary focus of this paper is to automatically identify collaborative (i.e. backhaul, load sharing) opportunities using pattern detection from the analysis of an extremely large-scale telematics dataset. To detect these opportunities a historical network model is developed by sequentially building up patterns automatically identified in the data. In sections 3, 4 and 5, frequently repeating patterns of trips are detected. In sections 6 and 7, we estimate the most likely routes taken by the frequent trips and embed numerous variables important for supply chain management. Using this historical network, backhaul is detected and ranked by its viability in section 8. In section 2, we provide a review of the current work related to telematics systems and backhaul detection.

2. RELATED WORK

In recent years, there has been a growing interest in collaborative logistics in the freight transportation and trucking industry. Traditionally, shippers and carriers have managed their operations independently. The new trend is to collaborate with other shippers and carriers, identify potential opportunities on the system level, and share the benefit of integrated operation costs among partners. Based on the IT architecture, there are two types of collaborative transportation systems. One type is a centralized, on-demand transportation management systems, including commercial service providers such as Nistevo, Transplace and LeanLogistics. Another type is adecentralized, online brokerage systems, such as 123loadboard and LoadAuthority. These systems provide a marketplace for private transportation participants, especially small-to-medium sized carriers, to exchange information, respond to demand and supply fluctuations, and optimize operation costs.

Prior work in the area of backhaul analysis has focused mainly on two major research areas. The first area involves economic models of backhaul opportunities. These studies include the determination of backhaul equilibriums, pricing models, and search time tradeoffs [3]. Other work focuses on the economic analysis of backhaul operations and the effect of regulation [4], and some analyses target the “uncertainties and complexities of scheduling” along with the effects of an auction [5]. The second area of research is on backhaul optimization models and algorithms for the vehicle routing problem (VRP). The vehicle routing problem is typically addressed using linear programming with constrains like demands (loads), pickup locations, delivery locations, time window constraints, fleet size, traveling time and other factors [6]. Heuristic approaches such as genetic algorithms (GA) or ant colony optimization (ACO) are also used to obtain fast and approximate solutions to the problem. [7]. However, in real world systems, these optimizations may take unacceptable computation time due to the large scale and dynamic nature of the network involved.

In order to enable near real-time detection of backhaul opportunities, we are studying the historical data obtained from a large-scale asset-tracking telematics system. [8] has proposed using data mining tools to learn dispatch rules from historical data, but their focus is not on geo-based telematics data. To the best of the authors’ knowledge, no study has been done to identify backhaul and collaboration opportunities using telematics data on a large-scale dataset spanning an entire continent.

To detect potential backhaul opportunities, frequent trips must first be mined from telematics data in order to determine patterns in freight movement. Trip extraction from automatically collected, telematics-based, GPS datasets has seen application in areas such as trip surveys, personal routine learning, route prediction, trip arrival time prediction, and transportation mode learning [9-14]. However, prior work on mining trips in telematics datasets has been performed mainly on time-based GPS data, in which data is collected at a predefined time frequency (e.g. 30 seconds, or 1 minute). The event-based GPS data addressed in this work are much more sparse compared to time-based GPS data. A trip can have as few as two data points indicating the trip-start event and the trip-end event, as opposed to tens or hundreds of data points in time-based GPS data scenarios. Thus, algorithms that apply to trip inference and extraction from time-based GPS data are not directly applicable to our case.

3. TELEMATICS DATA COLLECTION

In satellite or cellular-based telematics systems for asset tracking, a mobile asset, such as a trailer, is equipped with a tracking device that has an embedded GPS receiver as well as other sensors, and this tracking device communicates with a central data server via satellite or cellular communication. Location data, vehicle identification information, and timestamps are usually embedded in every telematics message. Due to engineering considerations such as power conservation in the field and limitations associated with communications cost, it is common to have assets send telematics messages from the tracking system only in an event-based messaging mode. Events represent either normal activities of an asset, such as “trip start”, “trip end”, “door open”, “door close”, “cargo loaded”, and “cargo empty”; or exceptions such as “lost GPS signal”, “unit low battery”, etc. In our system, each message generated by the tracking device contains one of these event codes and is annotated with geo-location and timestamp information, indicating where and when the activity or exception happened.

4. TRIP EXTRACTION

Definition: A telematics message in our event-based GPS dataset is defined as $P = \{lat, lon, t, e\}$, where *lat* is latitude, *lon* is longitude, t is timestamp and e is the event code. Event codes, e, can be from the set {"trip start", "trip end", "door open", "door close", "cargo loaded", "cargo empty", "lost GPS signal", "low battery"}.

The GPS data stream from a trailer is $\{P_1, P_2,..,P_n\}$, as shown in Figure 1.

	Latitude	Longitude	Time	Event
P_1	lat_1	lon_1	t_1	e_1
P_2	lat_2	lon_2	t_2	e_2
…	…	…	…	…
P_n	lat_n	lon_n	t_n	e_n

Figure 1. GPS data stream.

In our system, the tracking device, on the trailer section of a class 8 truck, automatically sends a "trip start" event message when the trailer starts a trip, and a "trip end" event when the trailer stops moving. Therefore, a basic rule for extracting trips is to represent a trip by the message sequences between consecutive "trip start" events and "trip end" events, as shown in Figure 2.

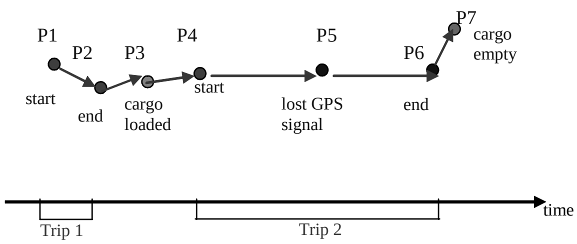

Figure 2. Trip segments.

Definition: A Trip $T= \{P_i, (P_{i+1}, \ldots),P_j\}$, where P_i and P_j are the consecutive "trip start" events and "trip end" events, and $(P_{i+1,\ldots,})$ are possible intermediate non-trip-related event messages.

Intermediate messages are very important to differentiate trips. Given the sparse nature of our data, most of the trip data will have only start and end points, with no information about which route the asset takes on the trip. Locations of those intermediate messages provide additional information about a trip and can be useful to differentiate trips with different routes.

Due to the noisy nature of our data, we observed several exceptions during the trip extraction process, as listed below:

- Either a "trip start" message or a "trip end" message is missing: Ideally, a "trip start" message and its corresponding "trip end" message should come in pairs and that would define a trip. However, in actual operation, "trip start" or "trip end" messages are sometimes missing. A "trip start" message may be followed by another "trip start" message, or a "trip end" message may not have its corresponding "trip start" message. We chose to filter out this type of incomplete trip information.
- In special cases, there could be multiple "trip start" messages or "trip end" messages missing. This can be inferred by checking if the time duration between a "trip start" message and its consequent "trip end" message exceeds a certain time threshold, such as 3 days. We chose to filter out this type of misinterpreted trip.
- A "trip start" message and its consequent "trip end" message are sent from the same location: in most cases this happens when an asset carried out a short distance roundtrip. If there are no intermediate messages, it is difficult to determine where the asset has been. We chose to filter out this type of trip.

The trip extraction algorithm based on the above heuristic rules is described below. The algorithm first retrieves GPS data streams for all assets. Then, it sequentially processes the GPS data stream, looking for consecutive "trip start" and "trip end" message pairs. The duration and distance between the consecutive "trip start" and "trip end" message pairs are calculated to filter out exceptional cases. The time threshold "δ" and spatial distance threshold "d" used in the algorithm are chosen heuristically and can be adjusted.

Algorithm: Trip Extraction

For each trailer, retrieve its GPS data stream $\{P_1, P_2,..,P_n\}$

For each message P_i in $\{P_1, P_2,..,P_n\}$

    If P_i is a "trip start" message

    $P_{start} = Pi$ % mark trip start

    Else if P_i is a "trip end" message and P_{start}! =Null

        If $|t_{start} - t_i| < \delta$ and $|(lat_{start}, lon_{start}), (lat_i, lon_i)| > d$

            Generate a trip record $T =\{P_{start}, (\ldots),P_i\}$

        End

        P_{start} =Null % reset trip start

    End

End

We tested our trip extraction algorithm on the telematics dataset available to us. We extracted 0.6 million trips from about 10 million raw GPS messages (collected over a 4-month period from more than 20,000 assets belonging to multiple fleets). Figure 3A is a visualization of a subset of the raw GPS messages. Figure 3B shows the trips of an individual trailer extracted from this subset. Figure 3C is a collection of trips by multiple trailers extracted from this GPS dataset.

Figure 3. Visualization of (A) raw GPS data, (B) trips extracted -individual trailer, and (C) trips extracted - multiple trailers.

5. TRIP ANALYSIS

In this section, we briefly describe our method to discover frequent trips in the derived trip datasets. In Figure 3C, we can see that, given a large number of trips, it is difficult to visually explore patterns. Trip clustering can be used to identify similar trips that have been repeated by multiple assets, multiple times. First we define "similarity" between trips. Given the sparse nature of our data, we only compare the start and the end locations of trips. Two trips are considered similar if their respective start and end locations are spatially close.

Definition: **(Similarity)** Two trips $T_1 = \{P_i^1, (P_{i+1}^1 \ldots), P_j^1\}$ and $T_2 = \{P_i^2, (P_{i+1}^2 \ldots), P_j^2\}$ are similar if $|((lat_i^1, lon_i^1), (lat_i^2, lon_i^2)| < d$ and $|((lat_j^1, lon_j^1), (lat_j^2, lon_j^2)| < d$, where d is the pre-determined distance criteria.

Depending on the accuracy of the GPS receiver and the physical size of the start or end location, different GPS coordinates may refer to the same location. For example, point 1 (42.3463, -71.0974), point 2 (42.3464, -71.0975), and point 3 (42.3460, -71.0976) all refer to the same location, "Fenway Park", in Boston, MA. Therefore, to determine whether the start or end locations of two trips are spatially similar, we need to use a small radius *d* to address the fuzziness of locations represented by GPS coordinates.

We used a modified k-means clustering algorithm similar to [9] to discover similar trips. In order to address the scale issue of the massive dataset, we added a grid-based indexing mechanism. First, all of the trips are grid-indexed based on their start and end locations. Then, during the clustering process, instead of including all of the trips in a pair-wise comparison for similarity calculations, only those trips with similar grid indices are considered. Experiments done on our large-scale real datasets prove that this algorithm can significantly cut down computation time while yielding accurate enough clustering results.

Figure 4, shows a supply chain network that emerges from trip analysis. The nodes represent significant locations such as distribution centers, stores, vendors, maintenance facilities, etc. The links are the frequent trips going between these locations and link thickness indicates trip frequency.

Figure 4. A supply chain network model created based on frequent trips.

6. FREQUENT ROUTES

Frequent trip data exists as a series of line segments connecting the different time-based, geo-referenced messages received from telematics devices. Frequently traveled routes can be estimated based on calculating the most likely route traveled to arrive at each vertex in the line segment. Line segments often grossly simplify the actual route taken. Line route estimate vs. a route prediction is shown in Figure 5.

Figure 5. Orange circles depict a temporally ordered sequence of telematics messages. Red lines show straight-line distance, while the green line depicts an estimation of the route taken.

An ordered series of three geo-coded messages is observed in the Figure. The straight-line distance is shown in red, while the estimated route is shown in green. The straight-line distance from location 1 to 2 is known to be very inaccurate because there are no major highways within a reasonable distance of the line. Restriction of trucks to major roadways further reduces the domain of possible routes traveled. The route shown in green is a likely estimation of the actual route.

To calculate the actual route, a geometric network is constructed using the standard API available in ArcGIS [15] software with the network analyst extension [16]. Developed by ESRI, this software allows for calculating routes and modeling the historical flow of monitored resources throughout a roadway network. Edges are weighted based on the cost or estimated travel-time to traverse each edge. The route with the minimal traversal time is often, but not always, considered the most likely traversed route. Additional information determined from analysis of telematics data can help validate the accuracy of the predicted route. The reasonableness of a proposed route is determined by comparing the estimated travel time to the actually observed duration. To further improve accuracy, intermediate messages are analyzed. These are event-based messages beyond start/end of trip information, such as "door open", or "cargo loaded." Since each frequent trip is comprised of a set of individual trips, the in-transit messages from each trip assists in determining the frequently traversed routes. The frequent routes thus derived are used to create a model of historical freight movement throughout North America.

7. DETERMINING HISTORICAL FREIGHT MOVEMENT

To create a model of freight movement, numerous variables need to be associated with the appropriate geographical locations and routes. These features include cargo status and frequency information. In our data set, cargo status is recorded as *1 = cargo, 0 = no cargo*. This is a Boolean condition; therefore there is no information about partial loads. Knowledge of cargo status is crucial in assessing backhaul opportunities - matching between empty trips and full trips occurring in the same direction. The frequency of these trips weights the backhaul opportunity in determining the likelihood that a collaborative match can occur within temporal restraints.

In our model, each road segment is embedded with cargo status and frequency information for each direction of travel. We define a variable, *cargo_status* as the ratio of full trips to total trips. For instance, if a route segment has a *cargo_status = 0*, all of the assets that traveled along that route were empty. Likewise, if the route segment has *cargo_status = 1*, then all of the assets were full. A *cargo_status* of *0.5* would indicate that half of the assets traveling this route segment were full, and half were empty. A similar process occurs for route frequency; the most substantial difference is that the frequency for each route is initialized based on the number of trips clustered together. To determine the cargo status and frequency at specific route segments, we need to combine routes that overlap.

In Figure 6, we see two routes. The green route depicts a trip hauling cargo, and the red route depicts an empty trip. Significant overlap exists, as indicated where these two routes intersect. At the intersection, the average cargo status is 50%. In other words, 50% of the trailers moving along the intersection have cargo. The frequency for the westwardly direction of the intersection is equal to *2*, since two routes overlap.

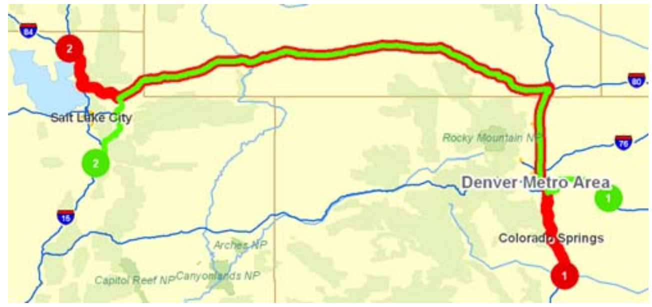

Figure 6. Two routes are shown. The red route shows a trip with an empty load; the green route shows a trip with a full load.

The calculation of these features is determined for all road segments. To determine *cargo_status* and *frequency* features, the roadway network must be segmented based on overlap and direction of travel. Since the values at the intersection are unique, a route segmentation algorithm creates a new route R_{new}, which is defined by the intersection of the green route (R_1) and the red route (R_2)

$$R_{new} = R_1 \cap R_2$$

Three routes are created from the two original routes to distinguish between the three different segments. The three segments are: (1) where the R_1 and R_2 intersect, the *frequency* is *2* and *cargo_status = 0.5*; (2) the disjoint set of R_1 and R_{new}, this defines the section of R_1 where R_1 and R_{new} do not overlap. In this case, *frequency = 1* and *cargo_status = 1* and (3) the disjoint set of R_2 and R_{new}, with *frequency = 1* and *cargo_status = 0*.

The different states that exist between the intersecting and non-intersecting segments require dividing two routes into three when a non-empty intersection occurs. This segmentation is performed for every route in the domain. If there is no intersection, then no sophisticated calculation is performed. Since all routes are compared, the compute time is quadratic or $O(n^2)$, where n is the total number of routes.

For each intersection, the frequency and cargo status is determined. Calculation of the frequency is straightforward. Initially, all the routes are initialized to equal the number of trips clustered to create the frequent trip. When two routes intersect, the frequency of the intersection equals the summation to the two frequencies. To calculate the *cargo_status*, an additional variable is used called *cargoTotal*, which is initialized by adding all the trips that have cargo. For each intersection, the value of *cargoTotal* is the summation of the two *cargoTotal* variables. To determine the *cargo_status* for the intersection, *cargoTotal* is divided by the *frequency*, after all the routes have been segmented.

Algorithm: Historical Network of Freight Movement

Add each frequent route to list *RouteList* $\{R_1, R_2,..,R_n\}$

For each i element in *RouteList*

  For each k element in *RouteList*

    R_i = *RouteList[i]*

    R_k = *RouteList[k]*

    If $R_i \cap R_k \neq \varnothing$ and (Same Direction) then

      Generate a new route R_{new}

      $R_{new} = R_i \cap R_k$

      $R_{new} \rightarrow frequency = R_i \rightarrow frequency + R_k \rightarrow frequency$

      $R_{new} \rightarrow cargoTotal = R_i \rightarrow cargoTotal + R_k \rightarrow cargoTotal$

      Add R_{new} to back of *RouteList*

      $R_i = R_i - R_k$

      $R_k = R_k - R_i$

    End

  End

End

For each j element in *RouteList*

  R_j = *RouteList[j]*

  $R_j \rightarrow cargo\_status = R_j \rightarrow cargoTotal \,/\, R_j \rightarrow frequency$

  Add R_j to the back of *RouteList*

End

$$Historical\_Network = \bigcup_{i=1}^{i=n} RouteList[i]$$

In Figure 7, sample results of the above algorithm are shown. This depicts the historical movement of goods through a supply chain. The colors from yellow to red show increasing frequency of routes, and the arrows illustrate the direction of travel. This supply chain was determined by merging all the routes in *RouteList*, into one network. This defines the historical movement of freight, essential in predicting backhaul opportunities.

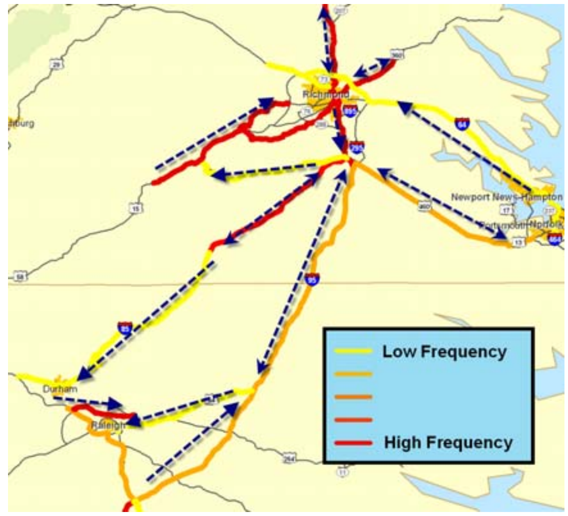

Figure 7. Historical pattern of high frequency freight movement.

8. DETECTING BACKHAUL OPPORTUNITIES

The historical freight movement network described in the previous section is computed once, and then updated as additional telematics information is collected. To detect backhaul opportunities, this historical network is queried based on a proposed empty trip. Therefore two inputs are required for determining backhaul collaboration opportunities: accurate determination of historical freight movement patterns and the current intentions of a distributor.

For example, consider a situation in which a driver has just delivered a shipment to the Virginia / North Carolina border. The driver now needs to return towards a distribution center outside the Richmond metropolitan area. The cost of fuel, driver's pay and vehicle overhead make it expensive to drive this long distance without pulling any revenue-generating cargo. A backhaul opportunity could exist if the empty trip intersects with a frequently traveled route. Our algorithm identities these potential intersections based on the historical freight movement patterns, as defined in the previous section, and the driver's current intentions.

At the intersection, the likelihood of collaboration is determined based on the frequency of the route and the percent of full cargo loads on that route (cargo_status). The higher the frequency, the more likely backhaul collaborations can occur within time restrictions, since waiting for potential loads is often expensive. Additionally, a high percentage of full loads is an indicator of a greater demand for hauling goods into a particular region [5]. Basic economic theory states that higher demand results in higher prices. Therefore the high demand for freight transportation in the same desired direction should result in substantial payoff and therefore increase the motivation for collaboration.

In Figure 8 we can see an illustration of the empty trip (green) from the Virginia / North Carolina border towards Richmond. The trip overlaps with a very highly traveled route in the same direction about halfway towards Richmond. Since both the frequency of trips and the percentage of full loads are high, there is a very significant opportunity to collaborate on backhaul. The purple route segment indicates where collaboration could occur.

Figure 8. The green route indicates a tractor-trailer returning to a distribution center after delivering a primary shipment. The trailer is pulling no cargo, but can likely collaborate with a third party to pick up a full load in the purple region.

Algorithm: Detect Backhaul

If *Historical_Network* $\bigcap$ *EmptyTrip* = $\varnothing$ then

```
     print "no viable backhaul opportunity"
Else
     RANK = frequency x cargo_status
End
```

The viability of the opportunity is determined by the frequency and average cargo status at the intersection of the empty trip and the historical freight network model. The algorithm for identifying and ranking these cases is shown above. If the empty trip does not intersect with a frequently traveled route, then historical trends indicate that it is more difficult to collaborate on backhaul. In these cases, visual inspection of the historical freight movement patterns can assist the fleet manager in identifying cost-saving behavioral changes. These changes may include deviations from the shortest route to allow for collaboration to occur.

The product of the frequency and cargo status (percentage full), determines a ranking of the viability of the backhaul. By identifying backhaul collaborative opportunities, the number of empty miles can be reduced, saving money on fuel, salary and vehicle costs and reducing emissions.

9. CONCLUSION - FUTURE WORK

In this paper we described a method for automatically detecting backhaul by building a historical model of freight movement from a large, real-world telematics network. Detection is based on intersecting a proposed empty trip with the historical network model and ranking the viability based on the frequency of a route and percentage of cargo full on that route. Future work will integrate risk factors that affect whether or not it is economically reasonable to take a backhaul opportunity. Better understanding of how external variables impact freight collaborations will assist in the development of services and tools that can better match loads across partners. Variables such as travel time reliability and network constraints (e.g. is a distribution center over capacity?) affect the risks associated with taking a particular backhaul. For example, a high variance in travel time correlates to higher risk. Better monitoring of risk factors can be incorporated into our model, allowing for a probabilistic ranking scheme of backhaul opportunities. These factors can also be used to develop an economic model that analyzes the risk factors and expenditures versus the payoff of taking a load. This will allow for better negotiations and decision-based algorithms for accepting viable matches. Future brokerage systems will automatically detect and rank these backhaul opportunities and provide multi-hop schedule recommendations, further reducing fleet costs and fuel consumption.

10. REFERENCES

[1] Standard Trucking and Transportation Statistics. American Trucking Association: Economics and Statistics Group. Vol 13, Issue 1. 2007.

[2] SmartWay EPA. 17th July 2008. U.S. Enviromental Protection Agency: Office of Transportation and Air Quality. http://www.epa.gov/smartway/

[3] Demirel E., Van Ommeren J., and Rietveld P. 2007. Matching Model for the Backhaul Problem. Tinbergen Institute Discussion Paper No. TI 07-083/3.

[4] Felton J. 1981. The Impact of Rate Regulation Upon ICC-Regulated Truck Back Hauls. Journal of Transport Economics and Policy 15, (3), pp. 253-267, 1981.

[5] Brewer P. and Plott C. 2002. A Decentralized, Smart Market Solution to a Class of Back-Haul Transportation Problems: Concepts and Experimental Test Beds. INTERFACES, Vol. 32, No. 5, pp. 13-36, 2002.

[6] Wang X. and Regan A. C. 2002. Local Truckload Pickup and Delivery with Hard Time Window Constraints. Transportation Research Part B. Vol. 36, pp. 78-94, 2002.

[7] Potvin, J. Y., Duhamel C. and Guertin F. 2003. A Genetic Algorithm For the Vehicle Routing Problem. Computer & Operations Research, Vol. 30, pp. 787-800, 2003.

[8] Chen W., Pi L., Shi L., Thomas B., Wikum E., Jarosinski G., Sun P., and Gifford T. 2007. Data Mining-Based Decision System Framework for Local Pickup and Delivery Problem, INFORMS Annual Meeting, 2007.

[9] Ashbrook D., and Starner T. 2002. Learning Significant Locations and Predicting User Movement with GPS. In Proceedings of the 6th International Symposium on Wearable Computers.

[10] Zhou C., Terveen L., and Shekhar S. 2005. Discovering Personal Paths from Sparse GPS Traces. In Proceedings of the JCIS 2005 Workshop on Data Mining, 2005 (WDM '05).

[11] Wolf J., Guensler R., and Bachman W. 2001. The Elimination of the Travel Diary: An Experiment to Derive Trip Purpose from GPS Travel Data. Annual Meeting of the Transportation Research Board, Washington, D.C.

[12] Liao L., Patterson D., Dieter F., and Kautz, H. 2006. Building Personal Maps From GPS Data. Annals of the New York Academy of Sciences, vol. 1093, pp. 249-265.

[13] Zhang Y., Liu L., Wang L., and Xie X. 2008. Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web. In Proceedings of World Wide Web Conference 2008. Beijing, China.

[14] Hudson M. and Rhys-Tyler G. 2004. Using GPS Data to Calculate the Length and Variability of Freight Vehicle Journey Times on Motorways. In Proceedings of 12th IEE International Conference on Road Transport Information and Control (RTIC 2004), pp. 49 – 55, 2004.

[15] Ormsby T., Napoleon E., Burke R., Groessl C., Feaster L. 2004. Getting to Know ArcGIS Desktop: Basics of ArcView, ArcEditor and ArcInfo. ESRI Press. Redlands, California.

[16] ArcView Network Analyst: An ESRI White Paper (1998) Environmental Systems Research Institute, Inc. Redlands, California.