HYBRID RULE BASED ARCHITECTURE FOR ORIGIN/DESTINATION EXTRACTION FROM ELECTRONIC FARE COLLECTION DATA

Markus Hofmann, Centre for Transport Research, Department of Civil, Structural and Environmental Engineering, Trinity College, Dublin 2. email: mhofmann@tcd.ie

Abstract

Obtaining Origin/Destination (OD) matrices from a public transport network is normally a work intensive and expensive procedure. OD matrices are mostly generated from data collected through travel diary surveys or traffic counts which usually focus only on a small area/sample. These methods of acquiring OD data may suffer from low response rates, response bias and tend to be time consuming and costly.

The aim of this paper is to present a method that facilitates the extraction of OD pairs of public transport passengers using entry validation electronic fare collection (EFC) data. The data used consist of approximately 47 million boarding records from a large Irish urban bus operator over a period of 2 years. The data are stored in an Oracle database. This paper reports on an algorithm that can successfully extract the destination attribute of single and transfer journeys. The assumption that the boarding of the second journey was the destination of the first journey is used. The algorithm can be applied to any urban bus operator with a similar data structure who has an EFC system with entry-only validation. A hybrid rule based and classification algorithm decides whether the passenger journeys are valid OD pairs or not using attributes such as unique ticket ID, route, direction, boarding time, boarding area and boarding zone.

This algorithm is a novel technique used to extract OD information which is then used to enrich the EFC database. This facilitates further analyses including parameters such as period of time, ticket type and ticketing category. OD matrices can be compiled from the database to serve as input for other public transport modelling software. The paper also shows possible analyses based on the extracted OD matrices.

Introduction & Background

Origin/Destination (OD) data is used in many transport models as one of the main parameters. The most common method of generating OD datasets is through surveys which are often expensive and time consuming (DFT-UK, 2004). Identifying possible sources of existing movement data can cause duplication of effort and further increases to the cost of transport modelling (DFT-UK, 2001). Consultants are particularly interested in timely sources of existing movement data due to the need of assessing alternative options when developing transport models (DFT-UK, 2001, 2004).

OD information is important for various analyses and planning purposes. However, boarding records stored by an entry only electronic fare collection (EFC) system do not reveal the location and time of alighting. The main focus of the research project on which this paper reports is to extract more detailed information about the location of alighting mainly with regard to spatial zones and areas of alighting as exact geographic coordinates were unavailable. The proposed algorithm aims to estimate the location of alighting by comparing boarding records from individual passengers which can be uniquely identified using the Ticket ID attribute. This information will then be extended with the OD type (36 different types) of the OD pair which contributes to the level of representativeness of the OD pair. The following two independent assumptions may lead to the successful implementation of an algorithm that can extract OD pairs:

1. Network symmetry may be assumed which means that each journey in one direction will also take place in the opposite direction (over a one day period) (Hofmann and O'Mahony, 2005).
2. The location of boarding in the evening can be considered the final destination in the morning and therefore facilitates the extraction of a valid OD pair (Furth, 2000; Richardson, 2003).

Although the first assumption can be used to obtain general OD matrices the execution is not based entirely on factual boarding records. It simply assumes that each journey also has a return journey with a certain error term. What time or which route was used can not be determined. See Hofmann and O'Mahony (2005) for further details on this assumed symmetry.

The second assumption uses the *Ticket ID* and *Ticket Type ID which* create a unique identifier for each passenger over a certain period of time (depending on the ticket type). This

information can be used to explore the path of the passenger and will be used as input data for the OD extraction algorithm. A travel diary is the entire set of journeys of one individual passenger throughout the validity period of the ticket. The algorithm has to incorporate patterns and travel records that at first do not result in a valid OD pair due to the complexity of the travel diary.

The assumptions used in this paper are similar to the assumptions used in (Barry *et al.*, 2002). Barry *et al.* (2002) implemented a similar concept on the data collected by the EFC system of the New York Metro. There are however many structural differences with regard to available data and complexity between Metro networks and bus networks. Nevertheless Barry *et al.* (2002) provided evidence that the assumptions are representative and mirror the results obtained by cordon counts.

Figure 1 shows the various stages of the algorithm and its implementation. The extension of the database and the data preparation ensures that location parameters of bus stops exist and that the data structures meet the requirements of the algorithm. The detection of subroutes is important to identify OD routes where passengers used a different route for the return journey. This will be introduced in greater detail throughout this paper. The identification of the OD scenarios provides a valid list which defines whether the passenger's boarding records actually build an OD pair or not. The OD extraction of single and transfer journey is carried out by the rule based algorithm. The results are then validated and imported into the existing EFC database.

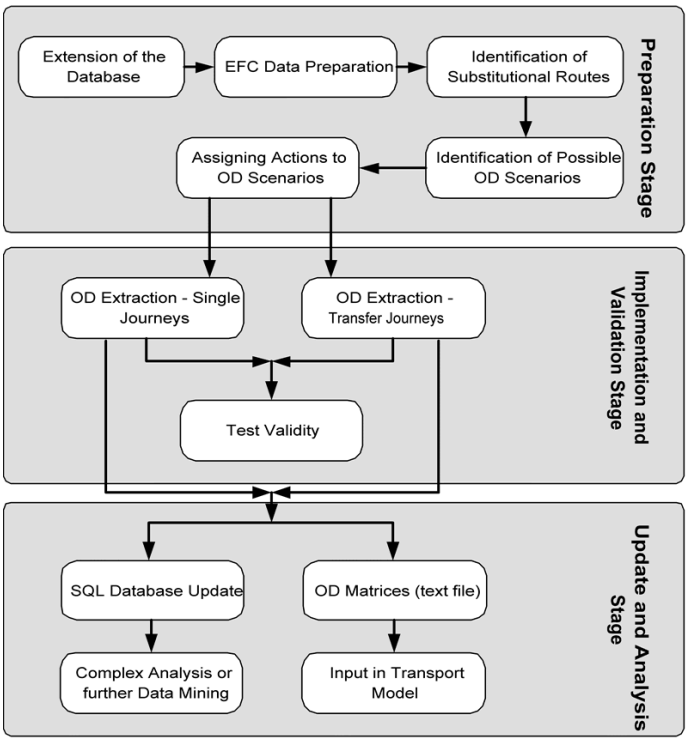

Figure 1: Structure of the OD extraction algorithm

The Assumption in Greater Detail

This paper focuses on the assumption that the boarding in the evening was the final destination in the morning. Figure 2 shows a diagram that simplifies the assumption for single journeys.

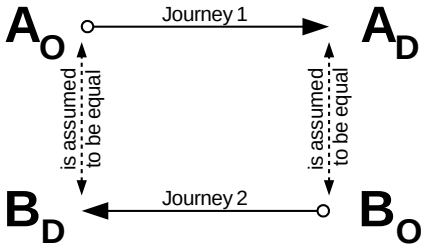

Figure 2: Visualisation of the Assumption for Single Journeys

A_O is the origin of journey 1, A_D is the destination of journey 1, B_O is the origin of journey 2 and B_D is the destination of journey 2. The assumption states that $B_O = A_D$ and $A_O = B_D$. Since A_O and B_O are the two known locations and $B_O = A_D$ and $A_O = B_D$ all locations (A_O, A_D, B_O and B_D) can be identified. Therefore two OD pairs can be extracted: A_O/B_O and B_O/A_O.

This referred to a single journey. Transfer journeys on the other hand add to the complexity of travel diaries extracted from EFC data due to their two legs that were recorded (this paper only includes transfer journeys with one transfer point). This means that four boardings were recorded to represent a return transfer journey. Figure 3 shows the stages of a one-transfer return journey consisting of 4 individual boardings (two for each journey – decoded as A, B, C and D). A_D/B_O and C_D/D_O are the transfer nodes where the passenger changed the bus in order to reach a final destination. Although A_D and B_O may have slightly different locations they do not influence the OD pair which aim it is to extract. It is assumed that the differences in location of B_O and A_D are within walking distance (same applies to D_O and C_D). Data of exact geographic coordinates of each boarding were not available for this project and it is therefore impossible to determine the exact distance between B_O and A_D or D_O and C_D. A_O, B_O, C_O and D_O are the known locations. The aim is to primarily determine the locations of A_D and D_D as this represents the main OD pair. The assumption states that $A_O = D_D$, $B_O = C_D$, $D_O = A_D$ and $C_O = A_D$ which results in the OD pairs A_O/D_O, B_O/C_O, C_O/B_O and D_O/A_O. Subsequently the OD pairs of interest are A_O/C_O and C_O/A_O.

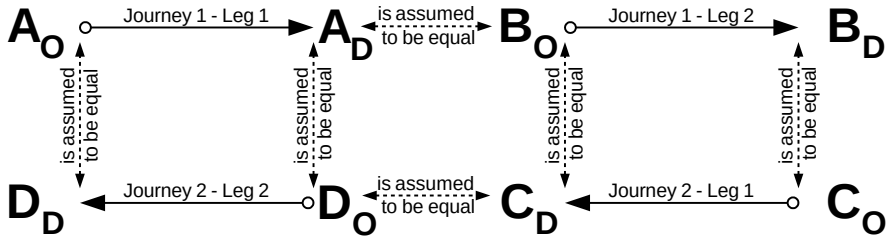

Figure 3: Visualisation of the Assumption for Transfer Journeys

The iterative classification algorithm described in (Hofmann *and O'Mahony*, 2004) was responsible to identify whether a boarding was part of a transfer journey or not. It is noteworthy to emphasise that not all consecutive journeys are return journeys. And not all journeys in the morning find their return journey in the evening. It is important to differentiate between single and transfer journeys as their data structure is different (2 boardings for single journeys and 4 boardings for transfer journeys). This difference also becomes apparent when the various travel scenarios are explored.

However, the above explained assumption is rather vague. The algorithm therefore needs to take more detailed information into consideration such as the bus routes the passenger chose for the return journey, the direction of the journeys, whether the OD pair was repeated throughout the ticket period and whether the boardings occurred in timely sequential order. All these parameters will be explained in the forthcoming sections.

The Data Source

The data source consists of the EFC data which was imported into a database (Hofmann *et al.*, 2003). The database facilitates the extraction of subsets of data. For performance improvement reasons with regard to run-time of the algorithm it was decided to extract data of one month as text files. The month the algorithm uses as data source is October 1999. This month was chosen as it was considered to have the least amount of bias which might have been caused by holidays or other seasonal adverse affects. The database stores 2.3 million boardings recorded during October 1999. Figure 4 shows a sample set of rows. *Boarding ID* is an internal number that serves the unique identification of each of the 46 million boardings stored in the database. *Ticket Type ID* is a three digit code that represents the type of a ticket such as *'Monthly Adult'* or *'Weekly Student'*. *Ticket ID* is a unique number that is assigned to each ticket and therefore serves as identification of each passenger. Figure 4 shows boarding records of two different passengers. Once the ticket ID is 3276 and once it is 3727. *Journey Type* represents the identifier whether the boarding was part of a transfer journey or not. '0' represents a transfer journey and '1' represents a single journey. *Date of Boarding* specifies the date the boarding took place. *Stage of Boarding* is a two-digit number that represents the bus stop on a particular route. *Time of Boarding* represents the actual time the passenger boarded the bus. *Route* represents the number of the route. *Direction* represents the direction of the route ('0' is outbound and '1' is inbound). *Coarse Zone* represents aggregated traffic zones. 21 different zones exist for the greater city zone. *Area Description* represents 132 different areas consisting of the names of city centre and urban area names (For more details see Hofmann and O'Mahony (2004)).

Figure 4 also indicates boarding records which the OD extraction algorithm can identify as OD pair. The highlighted single journey records build a valid OD pair in case the routes a9 and a5 are substitutional. The highlighted transfer journey example shows four boarding records which also build an OD pair. The passenger travels in the morning from Area 1 to Area 3 transferring in the City Centre South. In this instance the passenger used identical routes and the algorithm does not need to check for substitutional routes. However, as the bus network consists of many routes that follow the same path for at least some parts of the route it is important that the potential OD pairs are checked for substitutional routes to capture all OD pairs. The method used to determine substitutional routes is explained in the following section.

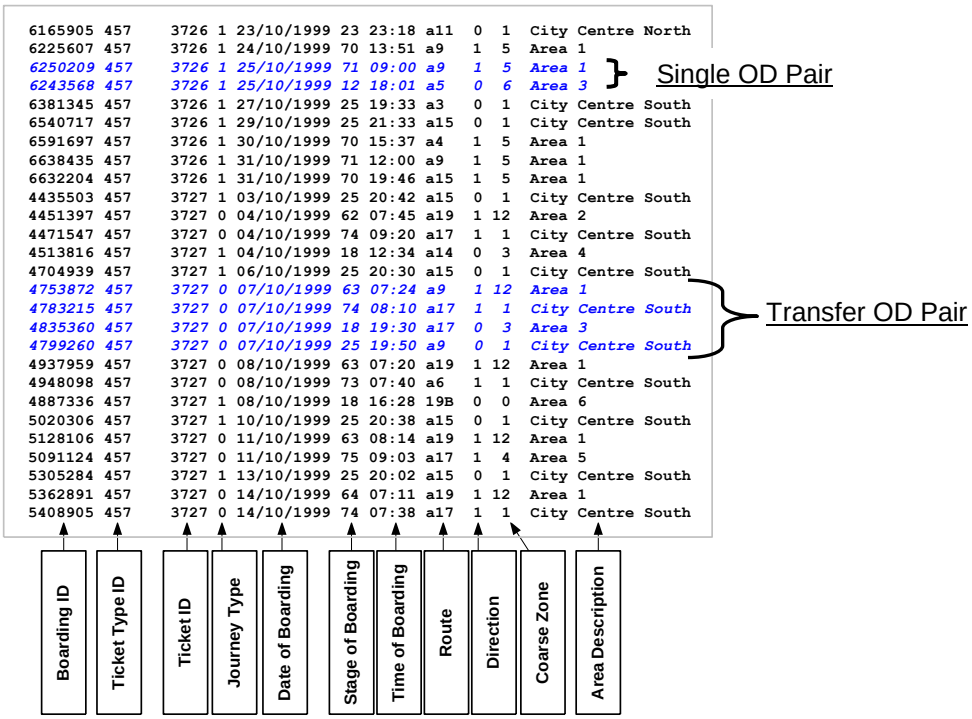

Figure 4: Data Input Format for the OD Algorithm

Substitutional Routes

Each alternative route a passenger can use to get to his/her final destination can be termed as *substitutional route*. Analysing the example of the potential single OD pair (see Figure 4) the boarding records only build an OD pair in case the routes a9 and a5 are substitutional. It is therefore important to understand which route is substitutional and which are the routes one particular route can be substituted with. It is vital for the successful implementation of the OD extraction algorithm that substitutional routes are incorporated. 40% of all extracted single OD pairs and 38% of all extracted transfer OD pairs show that passengers used a substitutional route for their return journey. The following explains the algorithm that was applied in defining substitutional routes:

1. The algorithm is based on the premise that substitutional route combinations are used in greater frequency than non-substitutional route combinations. The substitutional route identification algorithm (SRIA) generates a frequency table which contains a count of all route combinations that were recorded from single journeys that took place on the same day and in different direction. These pairs represent potential OD pairs if, and only if the routes are substitutional. Only boarding records that were in sequential consecutive order are considered. The sample data shown in Table 1 displays only a small subset of the frequencies of route combinations. It shows the route combination consisting of Route 1 (R1) and Route 2 (R2), the number of occurrences and the calculated weight of the route combinations which will be introduced in point 3. The data used as learning input consisted of boarding records collected throughout the month October 1999. The number of occurrences is a count of potential OD pairs with the according route combinations. For example, the SRIA identified that 776 passengers carried out a potential return journey using the route combination 66 and 66A.

Table 1: Sample data from the substitutional route list

Route 1	Route 2	Occurrences	Weight
66	66A	776	20.83
66	66	3,182	50.00
66	39	67	0.49
66	66B	749	20.78
66	67A	1,300	29.65
66	38	17	0.31
66	15	17	0.34
66A	66A	543	50.00
etc...	etc...	...	...
Total of 66	-	11,680	-
Total of 66A	-	3,377	-

2. The frequency table contains 6,645 different route combination pairs (such as 66/39 or 66A/66A) which are used to calculate the sums of all journeys from each route. 353,654 potential passenger combination pairs were identified.
3. Two parameters will be the decisive factors in identifying substitutional routes; the number of occurrences and a calculated weight. The number of occurrences of a particular route combination pair is not a satisfactory decision factor for defining substitutional routes because it would penalise small and less frequently used routes as their total passenger numbers would be smaller. The aim was therefore to include a second decision parameter which helps to successfully determine substitutional pairs focusing on the reduction of prediction errors. This lead to the development of an equation which represents a weighted indicator that determines together with the number of occurrences whether the route combination was in fact a substitutional route or not. The weight is calculated by dividing the total number of occurrences of one particular route combination (e.g. R_{66} and R_{66A}) by the total number of occurrences of $R_{66/66}$ plus the total number of occurrences of $R_{66A/66A}$. $R_{66/66}$ and $R_{66A/66A}$ are both the ideal route combination pairs as the return journey was not taken by a substitutional route.

$$S\ (R_{ij}) = \frac{T_{R_{ij}}}{T_{R_{ii}} + T_{R_{jj}}} * 100 \qquad \textbf{(1)}$$

where $S(R_{ij})$ is the calculated weight of the route combination R_{ij}. TR_{ii} and T_{Rjj} are the total number of occurrences of R_{ii} and R_{jj} respectively. T_{Rij} is the total number of occurrences of the particular route combination pair. Using the data from Table 1 the relative percentage of the combination 66/66A is therefore calculated as follows:

$$S(R_{66/66A}) = \frac{776}{3{,}182 + 543} * 100 = 20.83$$

This relative weighted value is calculated for all identified combination pairs. The range of possible values is $S(R_{ij}) > 0$. A frequency analysis of the calculated weights showed that 99.8% of all calculated values are below or equal to 50. A value of 50 is achieved when the weight is calculated for the same routes (i.e. $S(R_{66A/66A})$). A value above 100 is calculated when the number of occurrences of a route combination R_{ij} is larger than the combined number of occurrences of TR_{ii} and T_{Rjj}. The largest calculated weight for the dataset of this dataset was 91.89. This was calculated as follows:

$$S(R_{16/16A}) = \frac{2{,}344}{1{,}836 + 715} * 100 = 91.89$$

In this case $T_{R16/16A}$ had a total number of occurrences of 2,344. The denominator consisted of $T_{R16/16}$ and $T_{R16A/16A}$ which had a total number of occurrences of 1,836 and 715 respectively. This means that the substitutional route combination $R_{16/16A}$ is more frequently used than the non-substitutional route combinations $R_{16/16}$ and $R_{16A/16A}$. Again, with less than 0.2% these cases are the exception.
4. After calculating each route combination weight it was necessary to determine what values of these parameters classify the route combination pair as substitutional routes. Two cut-off conditions were defined to identify the substitutional routes.
    - Minimum number of occurrences. This means that the number of occurrences of a particular route combination has to be greater than the flexible pre-defined value.

- The weight S calculated in step 3 has to be greater than the flexible pre-defined value.

Using a 95% confidence interval and allowing for 5% sample error a simple random sample of 363 route combinations was selected from a population of 6,645 route combinations. It was then manually identified whether the route combinations of the sample are substitutional routes or not by comparing the course of the routes on maps provided by the urban bus operator. These results were then compared with the algorithm prediction results. The aim was to identify the most robust cut-off points for number of occurrences and the weight. It was also aimed to minimise the error of prediction. The testing on the sample showed the two best cut-off points with an estimation error of less than 2% at a minimum number of occurrence of 50 and a calculated weight of 1.5 or greater (see Figure 5). These cut-off points were then applied to the entire frequency table (see Table 1) identifying the substitutional routes which full-fill the cut-off point requirements.

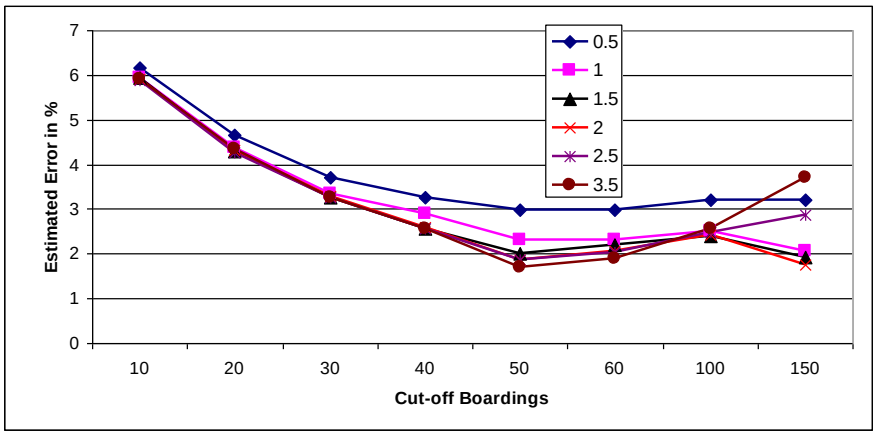

Figure 5: Impact of Errors on False Predictions

5. The final list is compiled by ignoring route combinations that are not classified as substitutional routes.

The Travel Scenarios

Travel scenarios are considered to be all physically and logically possible travel combinations that a passenger could have undertaken within the period of the ticket validity. The travel diary recorded for each customer reflects all journeys that were carried out by a particular passenger. These records have to be analysed by the OD extraction algorithm which then identifies the OD pairs for all logical boarding combinations that occurred. A travel scenario consists of two boarding records for single journeys and four boarding records for transfer journeys. A technique called *Decision Tables* is used to identify each of the possible travel scenarios. Decision tables are generally considered to be a set of variables or attributes which lead to an action, policy or alternative (Fernandez del Pozo *et al.*, 2003). Decision makers mostly use decision tables to search for the best recommendation for a certain case or attribute (Fernandez del Pozo *et al.*, 2003; Hewett and Leuchner, 2003). They are further used in knowledge-based decision support systems (DSS) as decision tables have the ability to represent complex logical relationships in an explicable and comprehensible manner (Cragun and Steudel, 1987; Slowinski, 1992; Kolodner, 1993; Hewett and Leuchner, 2003). For the purpose of this project, decision tables are mainly used to represent the logical relationships and as a programming methodology to support fast execution of the algorithm. A list of all structural patterns is identified which forms the foundation of the OD extraction algorithm. Structural patterns are used to structure data. This will also contribute to the semantics that are hidden in the data. For example, `IF Same Day AND Single Journey AND Different Direction AND Same Route AND Same Ticket Type AND Same Ticket ID THEN 'Positive OD Pair' ELSE 'No OD Pair'`. These structural patterns can be produced for all possible boarding record combinations as shown in the following paragraphs.
The following decision attributes are responsible for the categorisation of the scenario:
Date (D) ➔ The date is important with regard to the pattern that is represented by a passenger travel diary. The value for this attribute can either be 1 (same day) or 2 (different day).

Route (R) ➔ The route attribute focuses on the chosen routes of single journeys. Three values can be assigned to this decision attribute: 1 (both boardings took place on the same route), 2 (both boardings took place on a substitutional route) and 3 (the boardings took place on two different non-substitutional routes).
Route A/D (R A/D) ➔ The route attribute focuses on the chosen routes or boarding records A and D from a transfer journey record set (4 boarding records). Three values can be assigned to this decision attribute: 1 (both boardings took place on the same route), 2 (both boardings took place on a substitutional route) and 3 (the boardings took place on two different non-substitutional routes).
Route B/C (R B/C) ➔ The route attribute focuses on the chosen routes or boarding records B and C from a transfer journey record set (4 boarding records). Three values can be assigned to this decision attribute: 1 (both boardings took place on the same route), 2 (both boardings took place on a substitutional route) and 3 (the boardings took place on two different non-substitutional routes).
Direction (Dir) ➔ The direction attribute focuses on whether both boardings of single journeys have the same direction or not. Two values can be assigned to this decision attribute: 1 (the two boarding records have different directions) and 2 (both boardings have the same direction).
Direction A/D (Dir A/D) ➔ This direction attribute focuses on whether both boarding records of transfer journeys have the same direction or not. It compares boarding record A with boarding record D of a transfer journey record set (4 boardings). Two values can be assigned to this decision attribute: 1 (the two boarding records have different directions) and 2 (both boardings have the same direction).
Direction 2/3 (Dir 2/3) ➔ This direction attribute focuses on whether both boarding records of transfer journeys have the same direction or not. It compares boarding record B with boarding record C of a transfer journey record set (4 boardings). Two values can be assigned to this decision attribute: 1 (the two boarding records have different directions) and 2 (both boardings have the same direction).
Repeated (Rep) ➔ This attribute focuses on whether the OD pair was repeated with identical parameters such as route, stage and direction which is important with regard to the likelihood that the extracted OD pair actually occurred. Three values can be assigned to this attribute: 1 (the boarding pairs were repeated on the same day), 2 (the boarding pairs were repeated within the validity period of the ticket) and 3 (the boarding pairs were not repeated).
Journey (J) ➔ This attribute focuses on the type of journey. Two values can be assigned to this decision attribute: 1 (the boarding records represented a single journey) and 2 (the boarding records represented a transfer journey with one transfer).
Order (O) ➔ This attribute focuses on the order of the boarding records as they are compared. 1 (the boarding records are in consecutive order which means that they were recorded immediately after each other) and 2 (the boarding records were not in consecutive order).

Single Journeys
As mentioned above, two different decision tables will focus on identifying the logical alternatives out of all possible travel scenarios. This section focuses on the decision table for single journeys where only two individual boarding records are compared. Table 2 shows a list of all conditions and their values.

Table 2: Conditions and possible values for single journeys

Condition	Possible values of condition
Day	*(Same Day = 1, Following Day = 2)*
Route	*(Same Route = 1, Substitute Route = 2, Different Route = 3)*
Direction	*(Different direction = 1, Same Direction = 2)*
Repeated	*(Same day = 1, Period of ticket duration = 2, Not repeated = 3)*
Order	*(Consecutive = 1, Non-consecutive = 2)*

The number of theoretically possible scenarios is the product of all alternatives of each decision attribute (i.e. Day, Route, Direction, Repeated and Order). There are 2*3*2*3*2 = 72 possible combinations. As the only interest is to obtain valid OD pairs all the illogical combinations which do not lead to an OD pair can be omitted. The following elimination rules apply for single journeys leaving a total of 12 remaining combinations:

1. `IF Direction = {2} THEN Journey ≠ {Return Journey}` ➔ 36 eliminations
2. `IF Day = {2} THEN Journey ≠ {Return Journey}` ➔ 18 eliminations
3. `IF Route = 3 THEN Journey ≠ {Return Journey}` ➔ 6 eliminations

To (1): If the two boarding records have the same value for the direction attribute (i.e. outbound-outbound or inbound-inbound) it cannot be a return journey and therefore no valid OD pair can be obtained.
To (2): If the two boarding records took place on a different day then it cannot result in an OD pair.
To (3): If the two boarding records have a different value of the route attribute and were not identified as substitutional routes then no OD pair can be obtained.
Table 3 shows the 12 remaining combinations after eliminating the mutual exclusive and illogical combinations, the elimination of redundant conditions and the definition of the different scenarios for each combination. The example boarding records highlighted in Figure 4 (Single Journey) would be a Scenario 11 case (S11). As mentioned previously, knowing what type of OD pair was identified helps when analysing the representativeness of the OD matrix. The column 'Identified ODs' will be discussed in the results section.

Table 3: Structural patterns and rules for single journeys

Day	Route	Direction	Repeated	Order	Scenario	Identified ODs
Same Day	Same route	Different	Same day	Consecutive	S1	42,615
Same Day	Same route	Different	Same day	Non-consecutive	S2	3,158
Same Day	Same route	Different	Period of ticket	Consecutive	S3	92,924
Same Day	Same route	Different	Period of ticket	Non-consecutive	S4	8,625
Same Day	Same route	Different	Not repeated	Consecutive	S5	65,255
Same Day	Same route	Different	Not repeated	Non-consecutive	S6	3,100
Same Day	Sub route	Different	Same day	Consecutive	S7	8,317
Same Day	Sub route	Different	Same day	Non-consecutive	S8	1,300
Same Day	Sub route	Different	Period of ticket	Consecutive	S9	65,465
Same Day	Sub route	Different	Period of ticket	Non-consecutive	S10	8,230
Same Day	Sub route	Different	Not repeated	Consecutive	S11	70,644
Same Day	Sub route	Different	Not repeated	Non-consecutive	S12	7,347

Transfer Journeys
This section identifies all logical alternative travel scenarios for transfer journey records. Table 4 shows a list of all conditions and their values. The number of theoretical possible scenarios is the product of all alternatives of each decision attribute (i.e. Day, Route of A/D, Route of B/C, etc.). There are 2*3*3*2*2*3*2 = 432 possible combinations. As the only interest is to obtain valid OD pairs all the illogical combinations which do not lead to an OD pair can be omitted.

Table 4: Conditions and possible values for transfer journeys

Condition	Possible values of condition
Day	*(Same day = 1, Following day = 2)*
Route of A/D	*(Same Route = 1, Substitute Route = 2, Different Route = 3)*
Route of B/C	*(Same Route = 1, Substitute Route = 2, Different Route = 3)*
Direction of A/D	*(Different direction = 1, Same Direction = 2)*
Direction of B/C	*(Different direction = 1, Same Direction = 2)*
Repeated	*(Same day = 1, Period of ticket duration = 2, Not repeated = 3)*
Order	*(Consecutive = 1, Non-consecutive = 2)*

The following elimination rules apply for transfer journeys leaving a total of 40 possible remaining combinations:

1. `IF Direction (A/D) = {2} THEN Journey ≠ {Return Journey}` ➔ 216 eliminations
2. `IF Direction (B/C) = {2} THEN Journey ≠ {Return Journey}` ➔ 108 eliminations
3. `IF Day = {2} THEN Journey ≠ {Return Journey}` ➔ 54 eliminations
4. `IF Route (A/D) = {3} THEN Journey ≠ {Return Journey}` ➔ 18 eliminations
5. `IF Route (B/C) = {3} THEN Journey ≠ {Return Journey}` ➔ 12 eliminations

To (1): If the values of the direction attribute of transfer boarding records A and D are the same no OD pair can be obtained for this transfer journey.
To (2): If the values of the direction attribute of transfer boarding records B and C are the same no OD pair can be obtained for this transfer journey.
To (3): If the two boarding records took place on a different day then no OD pair can be obtained for this transfer journey.
To (4): If the values of the route attribute of journey records A and D are different (non-substitutional) then no OD pair can be obtained.
To (5): If the values of the route attribute of journey records B and C are different (non-substitutional) then no OD pair can be obtained.

Table 5 shows the 24 remaining combinations for transfer journeys after eliminating the mutual exclusive and illogical combinations, the elimination of redundant conditions and the definition of the different scenarios for each combination. The example boarding records highlighted in Figure 4 (transfer journey) would be a Scenario 5 case (T5). The column 'Identified ODs' will be discussed in the results section.

Table 5: Structural patterns and rules for transfer journeys

Day	Route A/D	Route B/C	Direction A/D	Direction B/C	Repeated	Order	Scenario	Identified OD's
Same	Same route	Same route	Different	Different	Same day	Consecutive	**T1**	45,201
Same	Same route	Same route	Different	Different	Same day	Non-consecutive	**T2**	3,315
Same	Same route	Same route	Different	Different	Period of ticket	Consecutive	**T3**	95,781
Same	Same route	Same route	Different	Different	Period of ticket	Non-consecutive	**T4**	8,841
Same	Same route	Same route	Different	Different	Not repeated	Consecutive	**T5**	70,328
Same	Same route	Same route	Different	Different	Not repeated	Non-consecutive	**T6**	3,443
Same	Sub route	Same route	Different	Different	Same day	Consecutive	**T7**	10,045
Same	Sub route	Same route	Different	Different	Same day	Non-consecutive	**T8**	1,457
Same	Sub route	Same route	Different	Different	Period of ticket	Consecutive	**T9**	66,748
Same	Sub route	Same route	Different	Different	Period of ticket	Non-consecutive	**T10**	8,360
Same	Sub route	Same route	Different	Different	Not repeated	Consecutive	**T11**	76,411
Same	Sub route	Same route	Different	Different	Not repeated	Non-consecutive	**T12**	8,056
Same	Same route	Sub route	Different	Different	Same day	Consecutive	**T13**	2,355
Same	Same route	Sub route	Different	Different	Same day	Non-consecutive	**T14**	388
Same	Same route	Sub route	Different	Different	Period of ticket	Consecutive	**T15**	2,815
Same	Same route	Sub route	Different	Different	Period of ticket	Non-consecutive	**T16**	461
Same	Same route	Sub route	Different	Different	Not repeated	Consecutive	**T17**	4,719
Same	Same route	Sub route	Different	Different	Not repeated	Non-consecutive	**T18**	871
Same	Sub route	Sub route	Different	Different	Same day	Consecutive	**T19**	1,862
Same	Sub route	Sub route	Different	Different	Same day	Non-consecutive	**T20**	529
Same	Sub route	Sub route	Different	Different	Period of ticket	Consecutive	**T21**	1,479
Same	Sub route	Sub route	Different	Different	Period of ticket	Non-consecutive	**T22**	475
Same	Sub route	Sub route	Different	Different	Not repeated	Consecutive	**T23**	6,843
Same	Sub route	Sub route	Different	Different	Not repeated	Non-consecutive	**T24**	2,211

Table 3 and Table 5 not only list the various travel scenarios but also serve as rules for the identification process. There are 12 rules for single journey OD identification and 24 rules for transfer journey OD identification. The next section introduces the technique Rule Based Reasoning which is used to extract the OD information.

Rule Based Reasoning

Rule Based Reasoning (RBR) is a data mining technique that uses a reasoning process to connect data to conclusions. It is mostly used in Expert Systems which are concerned with problem solving approaches (Ralston *et al.*, 2000). It is the formal definition of the thinking process when aiming to extract patterns. Production rules or simply rules are the most common method to represent knowledge (Ralston *et al.*, 2000). A rule is composed of a condition and an action - more commonly known as IF and THEN elements (Pedrycz, 2002). The IF element consists of a logical combination of conditions whereas the THEN element states the action which has to be taken in case the conditions are fulfilled. For example IF (certain conditions apply) THEN (take appropriate action). This type of knowledge representation is known as *action-oriented* (Ralston *et al.*, 2000). RBR are highly readable and have the advantage of being modular and highly modifiable which means that rules can be added and deleted in a flexible fashion (Pedrycz and Skowron, 2002). The rule base (also known as knowledge base) is the location where the identified rules are stored. In this project the rules are stored in two separate text files; one for single journeys and one for transfer journeys. The rules were identified in the Travel Scenario section (see Table 3 and Table 5) where decision tables and expert knowledge were used to define all possible and logical scenarios. The algorithm (as described in the following section) compares boarding records and extracts the information needed to identify the OD pair (i.e. route, direction, order, repeated, day) and then attempts to match the information with the rules stored in the rule base. If the gathered information matches a rule the algorithm found an OD pair and outputs this to a file. If no match is found the algorithm compares the next boarding record.

The following section connects all the previous sections and details the procedural structure of the algorithm.

Algorithm Development

This section describes how the entire algorithm is structured. The algorithm is programmed in C++ which input/outputs from and to various text files. The aim was to include expert knowledge into the algorithm which was achieved using the above described techniques. The following steps are carried out by the algorithm:

- Initially all necessary parameters have to be set. This includes the following:
  - Run substitutional route identification algorithm (Yes/No)
  - Recalculate substitutional route matrix with new cut-off values (Yes/No)
  - Set value for cut off value for number of occurrences
  - Set value for cut-off value for calculated weights
  - Run OD identification for single journeys (Yes/No)
  - Run OD identification for transfer journeys (Yes/No)
  - Output Single OD pair in 1 line (Yes/No)
  - Output Transfer OD pair in 1 line (Yes/No)
  - Output Single OD pairs directly into the Oracle database (Yes/No)
  - Output Transfer OD pairs directly into the Oracle database (Yes/No)
- If the 'Run substitutional route identification algorithm' is selected then the algorithm initially compiles a new list of substitutional routes. This procedure was described previously.
- If 'Recalculate substitutional route matrix with new cut-off values' is selected the algorithm re-calculates substitutional route list using the new values of the cut-off points.
- The first data file is opened
- All boarding records of one passenger are extracted
- The next part of the algorithm compares boarding records of single journeys. This contains the comparison of all single journey boarding records of each passenger. The comparison is done in pairs. A valid OD pair can only be extracted when the two boarding records were recorded on the same day. The following procedure is carried out for each comparison:
  - Route - if it is the same route then the route identifier value is 1. In case the routes are different it has to be tested whether they are substitutional routes. This is done by accessing the substitutional route file and comparing the two routes from the potential OD pair to the list of substitutional routes stored in the file. If the route combination pair is considered a substitutional route then the value is 2 otherwise the route identifier is 3.
  - Direction - Same direction of the two boardings of the potential OD pair results in a 1 and different direction in a 2 as values for the direction identifier.
  - Repeated - At this stage a positive OD pair has been identified and the algorithm's aim is to find same OD pairs of the same passenger throughout the validity period of the ticket. In case an OD pair is found repeatedly the repeat identifier changes. 1 for repeats on the same day, 2 for repeats during the ticket duration and 3 for not repeated are assigned. The substitutional route identification algorithm is also applied throughout the search for repeated occurrences of the OD pair. A further output is the actual number of repeats that were found.
  - Order - This part of the algorithm checks whether the boarding records which build the OD pair were in consecutive sequential order or not. 1 is assigned to consecutive order and 2 for non-consecutive boarding records.
- This part of the algorithm compares boarding records of transfer journeys. The characteristics of transfer OD pairs are different as 4 boarding records are required to find a valid OD pair. A valid OD pair of a transfer journey can only be extracted if all four boardings were recorded on the same day. After identifying 4 boarding records that potentially could form an OD pair its characteristics have to be identified. The following procedure is carried out for each comparison:
  - Route A/D - This compares the routes of boarding record A and D. The route has to be the same or substitutional. The Route A/D identifier is 1 for same routes, 2 for substitutional routes and 3 for different routes.
  - Route B/C - This compares the routes of boarding record B and C. The route has to be the same or substitutional. The Route B/C identifier is 1 for same routes, 2 for substitutional routes and 3 for different routes.
  - Direction A/D - This compares the direction of boarding record A and D. Same direction is a 1 and different direction is a 2.

  - Direction B/C - This compares the direction of boarding record B and C. Same direction is a 1 and different direction is a 2.
  - Repeated - At this stage a positive OD pair has been identified and the algorithm's aim is to find same OD pairs of the same passenger throughout the validity period of the ticket. In case an OD pair is found repeatedly the repeat identifier changes. 1 for repeats on the same day, 2 for repeats during the ticket duration and 3 for not repeated are assigned. The substitutional route identification algorithm is also applied throughout the search for repeated occurrences of the OD pair. A further output is the actual number of repeats that were found.
  - Order - This part of the algorithm checks whether all four boarding records, which build the OD pair, were in consecutive sequential order or not. 1 is assigned to consecutive order and 2 for non-consecutive boarding records.
- The result of the above described algorithm is a set of numbers that describe the found characteristics of the OD pair which now has to be matched to the rule base. Depending on the journey type (single or transfer) the algorithm accesses the rule base and attempts to match the extracted characteristics of the potential OD pair with one of the pre-defined rules (Table 3 and Table 5). Once a positive match is identified the algorithm outputs the boarding records as verified OD pair using the OD scenario assigned to the rule.
- If there are further boarding records in the file then the next set of boarding records of one particular passenger are extracted and the above described procedure is carried out again in the attempt to find a new OD pair. If no boarding records are left to analyse the next file is opened and the algorithm extracts the sets of boarding records from there.

The algorithm takes approximately 100 hours to analyse 2.3 million boarding records (one month). The algorithm was executed on a P4 with 2.4GHz and 1024MB of RAM.

Results & Potential Analyses

The OD information includes all the data attributes mentioned in Figure 4 for all records that were part of the return trip. Depending on the ticket type the algorithm was able to identify up to 79% of the boarding records as OD pairs. The success rate of the algorithm for weekly city tickets was 50% and 43% for monthly tickets. It is noteworthy to emphasis that not all journeys are return journeys due to car pooling or trip chaining, single trips. In total the algorithm identified over 422,994 OD pairs out of a total of 2.3 million boarding records (this number also includes single tickets and other tickets which don't facilitate the extraction of OD pairs).

Table 3 and Table 5 show the number of OD pairs identified for each travel scenario. The probability that the OD algorithm actually extracted correct OD pairs changes for each travel scenario. For example are travel scenarios which were repeated and in consecutive order more representative than scenarios which were not repeated and in non-consecutive order. 92% of all single journey OD pairs were recorded in consecutive timely sequential order. For transfer journeys this percentage was at 85.5% slightly lower. 61% of all identified single journey OD pairs and 42% of all transfer journey OD pairs were repeated either on the same day or throughout the period of ticket validity. The considerably lower percentage of the transfer journey OD repeats is probably caused by the added complexity of transfer journey records.

The extracted OD's can be analysed in many different ways using different methodologies. The focus of the analysis of OD pairs is very versatile. Analyses could be carried out for operational, strategical and behavioural aspects of the public transport sector. The operational use could be to analyse travel times of passengers on a system wide level focusing on different times of the day. The OD matrix could facilitate strategical analysis such as the implementation of new routes by analysing the path of passenger transfer journeys and thereby identifying the OD needs of passengers. The travel behaviour of public transport passengers is currently one of the main research interests. Knowing the OD of individual passengers their travel path for a certain period of time is known which may help to analyse their behaviour.

Each OD pair was fed back into the database and it is now possible to facilitate further analyses with regard to operational, strategically or behavioural focus.

Conclusion

This paper presented a novel algorithm that facilitates the extraction of OD pairs from already recorded EFC data from an urban bus operator. In transport networks where magnetic stripe cards or smart cards are widely used this approach could save time and money and would also facilitate analyses on a network wide level. After the extraction the data were fed back into the database to

facilitate a more complex analysis. Up to 79% of the boarding records were identified as OD pairs. The success rate of the algorithm for weekly city tickets was 50% and 43% for monthly tickets.

The paper presented a novel method to estimate substitutional routes on a network wide level. The introduced equation calculates a weight which in combination with the number of passenger's route combination occurrences serves as a determinant whether two routes are substitutional or not. Using a simple random sample the estimation error of the algorithm was less than 2%. The generic method can be applied to any system that provides the required EFC data.

Table 3 and Table 5 showed the number of OD pairs identified for each travel scenario. The probability that the OD algorithm actually extracted correct OD pairs changes for each travel scenario. 92% of all single journey OD pairs were recorded in consecutive timely sequential order. For transfer journeys this percentage was at 85.5% slightly lower. 61% of all identified single journey OD pairs and 42% of all transfer journey OD pairs were repeated either on the same day or throughout the period of ticket validity. The considerably lower percentage of the transfer journey OD repeats is probably caused by the added complexity of transfer journey records.

The newly generated information can be used for operational, strategical and behavioural analyses.

Acknowledgement

The research has been funded by the Department of Transport under the Transport Research Programme administered by the Higher Education Authority.

References

Barry, J. J., Newhouser, R. Rahbee, A. and S Sayeda (2002) Origin and Destination Estimation in New York City with Automated Fare System Data, Transportation Planning and Analysis, Transportation Research Record, Volume 1817, pp.183 – 187.

Cragun B. and H. Steudel (1987) A decision-table based processor for checking completeness and consistency in rule-based expert systems. *International Journal of Man-Machine Studies*, 5:633–648.

DFT-UK (2001) National origin-destination databank - database application: User manual - version 1.0. Technical report, Department for Transport, London, UK. URL http://www.dft.gov.uk/stellent/groups/dft transstrat/documents/page/dft transstrat504834.hcsp [13 April 2004].

DFT-UK (2004) National origin-destination transport survey. Online, URL http://www.dft.gov.uk/stellent/groups/dft transstrat/documents/pdf/dft transstrat pdf 504835.pdf. [13 April 2004].

Fernandez del Pozo, J. A., C. Bielza, and M Gmez (2003) A list-based compact representation for large decision tables management. *European Journal of Operational Research*, In Press, Corrected Proof, URL: http://www.sciencedirect.com/science/article/B6VCT-4B7HK9Y-1/2/364d3ab07d8ae0780a62d6b5037a06a7.

Furth, P. (2000) Data analysis for bus planning and monitoring. Synthesis of transit practice 34, Transit Cooperative Research Program - Transportation Research Board, Washington D.C., USA.

Hewett R. and J. Leuchner (2003) Restructuring decision tables for elucidation of knowledge. *Data & Knowledge Engineering*, 46(3):271–290.

Hofmann, M. and M. M O'Mahony (2003) A Framework to Utilise Urban Bus Data for Advanced Data Analysis, Proceedings of the 10th World Congress on Intelligent Transport Systems and Services, Madrid.

Hofmann, M. and M. M O'Mahony (2004) Development of an Iterative Classification Algorithm to Facilitate Transfer Journey Analyses, 36th Annual Conference of University Transport Study Group, Newcastle.

Hofmann, M. and M. O'Mahony (2005) An analysis of route symmetry in an urban bus public transport network using electronic fare collection data. *Transportation Research Record*, In Press.

Kolodner J. (1993) *Case-Based Reasoning*. Morgan Kaufmann.

Pedrycz W. and A. Skowron (2002) Fuzzy and rough sets. In W. Kloesgen, editor, *Handbook of Data Mining and Knowledge Discovery*, Chapter 30, pages 680–690. Oxford University Press, Inc.

Pedrycz, W. (2002) Fuzzy sets perspective on data and knowledge. In W. Kloesgen, editor, *Handbook of Data Mining and Knowledge Discovery*, Chapter 13, pages 150–169. Oxford University Press, Inc.

Ralston, A., E.D. Reilly, and D. (2000) Hemmendinger. *Encyclopaedia Of Computer Science*. Grove's Dictionaries, 4th edition.

Richardson A. J. (2003) Estimating average distance travelled from bus boarding counts. 82nd Annual Meeting of the Transportation Research Board, TRB National Research Council. (CD-ROM), Washington D.C., USA.

Slowinski, R. (1992) *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Set Theory*. Kluwer Academic Publishers.