Hybrid Heuristic for Multi-carrier Transportation Plans

Dario Landa-Silva[1], Yijun Wang[1,2], Paul Donovan[2], Graham Kendall[1]

[1] ASAP Research Group, School of Computer Science
University of Nottingham, United Kingdom
dario.landasilva@nottingham.ac.uk, graham.kendall@nottingham.ac.uk

[2] 3T Logistics Limited
5 Smith Way, LE19 1SX
Enderby, Leicestershire, United Kingdom
yijun.wang@3t-europe.com, paul.donovan@3t-europe.com

Abstract

This paper describes a hybrid heuristic approach to construct transportation plans for a single-customer multi-carrier scenario that arises at 3T Logistics Ltd, a UK company that provides outsourced transportation planning and management services. The problem consists on planning the delivery, using a set of carrier companies, of a set of shipments from a warehouse to different consignees across the UK. The problem tackled resembles a vehicle routing problem with time windows but there are several differences in our scenario. The hybrid heuristic algorithm described here combines a clustering algorithm, constructive and local search heuristics, and exact assignment based on integer programming. This approach is being currently evaluated at the company and results so far indicate the suitability of the algorithm to produce practical transportation plans at reduced cost compared to current practice.

1 Introduction

3T Logistics Limited (3TL) is a company based in the UK that provides outsourced transportation planning and management services to manufacturing and distribution companies through *Global Carrier Manager*, their bespoke multi-mode transportation management system. The company produces transportation plans for the collection and delivery of goods from customers to consignees through the use of a number of carrier companies.

Providing global and integrated logistic services [1, 4, 5] is the driving philosophy at 3TL. The company faces different types of scenarios when providing transportation planning for its customers. One of these scenarios is called here *single-customer multi-carrier transportation planning* and refers to loading vehicles with shipments, sequencing the order of delivery and assigning each load to a specific carrier company. This problem is related to the loading [2] and vehicle routing with time windows [6] problems. The solution approach presented in this paper is a hybrid heuristic that incorporates clustering, constructive heuristics, local search and integer programming. The method consists of several phases that follow the overall logic that human planners at 3TL use to construct the transportation plans, from identifying geographical regions to assigning vehicle loads to available carrier companies. Results so far show that the proposed algorithm constructs transportation plans at considerable reduced cost compared to those produced by the human experts. This approach is currently being deployed by 3TL within their *Globar Carrier Manager* software for its use in their live planning operations.

Section 2 describes in detail the transportation planning problem tackled in this paper. Section 3 describes the hybrid heuristic approach used to tackle the problem. Then, Section 4 presents the experiments and results conducted in order to assess the suitability of this method for 3T Logistics Ltd.

2 Single-customer Multi-carrier Transportation Planning

The single-customer multi-carrier planning scenario that we tackle here is described as follows. There is a single customer that needs to send a number of shipments to consignees around the country. A shipment is a unique order and depending on its size it is handled using a given number of pallets, this number is given as PSE (pallet space equivalent). All shipments leave from a single source (i.e. 3TL customer's warehouse) and are delivered by vehicles hired from a set of carrier companies. There are three types of shipments:

- FTL (Full Truck Load): these are 'large size' shipments with a volume of $\geq 40\,m^3$, transportation cost is calculated by vehicle (see below).
- LTL (Less Than a Truck Load): there are 'medium size' shipments with volume $< 40\,m^3$ and > 3 PSE (handling units), transportation cost is calculated by vehicle.
- Groupage: these are 'small size' shipments that consist of ≤ 3 PSE and $> 2\,m^3$, this basically means that these small shipments can also be delivered by a courier service company[1], transportation cost may be calculated by vehicle or by shipment.

Planning is done for each day and the number of shipments usually exceeds the available capacity on the vehicles from the various carriers. The *transportation mode* indicates the way in which a shipment is delivered and it refers mainly to how the transportation cost is calculated. The *load mode* is when one or more shipments are sent using a vehicle hired from a carrier company, cost is given by the hiring of the vehicle. A *load* is a set of shipments to be delivered with a vehicle hired from a carrier company. The *parcel mode* is when shipments are passed from 3TL to a courier service company to be delivered, cost is given by the weight of each individual shipment. That is, in *load mode* collection and delivery of a shipment is made by the same vehicle while in *parcel mode* this might not be the case. Usually, only *Groupage* shipments may be delivered either in *load mode* or *parcel mode*, LTL and FTL shipments are delivered in *load mode* only. For each shipment (FTL, LTL and Groupage) the following is given: size (weight, PSE, volume), destination (postcode), delivery time (specific time or flexible time window), and estimated transportation cost (for different transportation modes and different carriers). There is a profile of available carriers, given by the number of vehicles, their capacity, their cost structure and the regions/lanes serviced. There is usually a choice of carriers for the same delivery and therefore planning also involves selecting what carrier to use for a given load. Driving distance and time are given between source and each destination, and between destinations. There are loading/unloading times (processing time) given for sources and destinations.

The problem is to produce *transportation plans*, that is, to form loads (for *load mode* transportation) with shipments of different types and define the sequence of delivery for each load ensuring minimum transportation cost while satisfying various restrictions such as vehicle capacity, carriers availability, geographical shipment compatibility, delivery time windows and driving time conditions. Shipments that do not fit into loads are sent in *parcel mode* and then only the corresponding shipment cost is considered (i.e. no delivery time windows). The transportation plan indicates what shipments are assigned to each load, which carrier is assigned to each load and information about routes, driving times, delivery times, etc. Since vehicles are hired for delivering the load only, the last delivery point is considered the end of the route because carriers might assign that vehicle to another job (unrelated to our plans) that day or another day. It is important to avoid *backward mileage* which happens when a later delivery point is closer to the source than some of the previous delivery points. That is, carriers prefer to deliver progressively farer from the source because they define their cost structure based of the most distant delivery made.

The following restrictions are considered in this *single-customer multi-carrier transportation planning* scenario:

- FTL and LTL shipments must be planned in *load mode*, Groupage shipments may be planned in *load or parcel mode*.
- Vehicle capacity (80 m^3 here) may not be exceeded.

[1]Shipments with 2 m^3 or less exist but are not considered in this planning scenario.

- There is an upper limit on the number of vehicles per carrier available.
- Each vehicle should be used around 9 hours per day (maximum 12 but not preferred).
- Each vehicle starts from the source and ends at the last delivery point.
- Avoid backward mileage although a limited amount (e.g. 15 miles) might be acceptable.
- The same delivery point may be visited more than once by different vehicles delivering different shipments.
- Loading times at the source are flexible, so it is assumed that vehicles collect all shipments on time in order to depart to the first delivery.
- Unloading time for each shipment is considered to be 0.5 hours.
- Most delivery times are given as a time window (e.g. 9:00 to 16:30 or 9:00 to 12:00) but some are fixed (e.g. 15:00).
- Maximum 4 deliveries per load or trip are allowed (drivers preference) or 6 deliveries if involving subgroups. A subgroup is a set of delivery points that are very close (within 5 miles) between them.
- Distance between the delivery points of any two shipments in the same load should not be more than 70 miles (hence we use a clustering stage).

When assigning loads to carriers, the different transportation costs must be taken into account. Each carrier company offers their own cost structure but in general it includes the cost of hiring the vehicle (according to the farest delivery) plus the cost per extra delivery. A typical transportation plan would involve around 60 to 100 shipments being delivered using around 35 to 70 vehicles hired from 15 different carriers.

3 Hybrid Heuristic to Produce Transportation Plans

We have developed a hybrid heuristic procedure to produce transportation plans for the single-customer multi-carrier transportation scenario described above. This approach combines a clustering algorithm, constructive and local search heuristics, and exact assignment based on integer programming. The clustering stage identifies groups of shipments that are 'nearby' in some sense (more details below). The constructive and local search heuristics are used to combine shipments into loads and define the delivery sequence of each load. At present, since there are usually few shipments within a load, we tackle the sequencing of shipments at the same time as building the loads. The exact assignment stage assigns a specific carrier (vehicle) to each load seeking to minimise the overall transportation cost.

It should be noted that this staged approach was taken instead of an integrated optimisation approach, after considering practical issues expressed by human planners at 3TL. For example, one of these issues is that the set of available carriers varies with the geographical region and hence it helps to first define clusters and later assign carriers to loads formed within each cluster. Also, the algorithm described here is being used to assist the human planners and having a staged approach makes easier for them to assess the plans and intervene in the process when required. An outline of the steps in the hybrid heuristic algorithm is given below.

1. **Clustering Shipments:** take all shipments (FTLs, LTLs and Groupage) and organise them into clusters of geographically compatible shipments. The purpose is to automatically generate 'post-code delivery regions'. We use the DBScan clustering algorithm by Ester et al. [3].

2. **Create Subgroups:** within each cluster, form subgroups of no more than 3 shipments that are very close (5 miles or less) between them (only FTL and LTL are considered). These subgroups are treated as a single shipment later when building loads. The rationale for this is drivers' preferences but also because carriers might charge for the delivery of several very close sequential deliveries as a single delivery hence reducing the overall cost.

3. **Build Initial Loads:** build loads for each cluster, first assign FTL shipments followed by LTL and Groupage shipments to form feasible loads only considering the capacity of available vehicles. For each load, determine the delivery sequence and calculate the quality of the plan (delivery time violations, waiting times, etc.) so far. Note that strictly satisfying the delivery times is not enforced in this stage.

4. **Carrier Assignment:** assign each of the built loads to an available carrier (vehicle) with the objective of minimising the overall cost of the transportation plan. We solve an IP formulation of the assignment problem in this step. Calculate the quality of the plan produced so far considering the *load mode* and *parcel mode* delivery costs involved.

5. **Improve Loads:** for each cluster, improve the loads by trying to eliminate violations of delivery times and reducing overall delivery cost. For this, we use a set of heuristics to move shipments between loads and also to re-sequence the shipments within a load.

The next subsections give more details for each step of the hybrid heuristic approach outlined above.

3.1 Clustering Shipments

Clustering is required in order to dynamically identify regions across the UK. We generate clusters so that shipments in each cluster are *geographically compatible*, i.e. separated by no more than a given driving distance. Lets consider X_{ftl}, X_{ltl} and X_{grp} to be the sets of FTL, LTL and Groupage shipments respectively. We adapted the DBScan (Density-Based Spatial Clustering of Applications with Noise) algorithm by Ester et al. [3] to find out the number of clusters and the boundary of each cluster given the set of shipments $X_{ftl} \bigcup X_{ltl} \bigcup X_{grp}$.

DBScan requires two parameters: the neighbourhood threshold (ϵ) and the minimum number of points required to form a cluster (*minPts*). The algorithm starts with an arbitrary initial point not yet visited. This point's ϵ-neighborhood is retrieved, and if it contains at least *minPts* points, a cluster is started. Otherwise, the point is labeled as 'noise'. However, this 'noise' point might later be added to the ϵ-neighbourhood of another point to form a cluster. When a point is found to be part of a cluster, its ϵ-neighborhood is also part of that cluster. Hence, all points in the ϵ-neighborhood are recursively added to a cluster until the cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of another cluster or 'noise'.

We adapted DBScan by adding a routine to split 'too big' clusters, avoid 'too small' clusters and absorb 'noise points'. Too big clusters (more than 20 points) are split by executing DBScan on that cluster with a value of ϵ smaller to 70 miles. Too small clusters are avoided by adjusting *minPts*. Noise points within $\epsilon = 70$ miles of another clustered point are absorbed by assigning them to their nearest cluster and by executing DBScan on groups of noise points. That is, is a 'noise point' is farer that 70 miles from its closest clustered point, the 'noise point' stays as such.

3.2 Create Subgroups

FTL and LTL shipments that are very close to each other (5 miles or less) form a subgroup to consider them as a single delivery point when building loads and calculating carriers cost. When forming subgroups we ensure that the vehicle capacity is not exceeded, i.e. the subgroup added volume is no more than 80 m^3. Also, the subgroup delivery times must be 'compatible', i.e. delivery times of all shipments in the subgroup can be met by the same vehicle. The number of shipments in a subgroup is limited (to 3 in this paper) due to operational reasons. We use the nearest neighbour heuristic [6] to form subgroups while enforcing vehicle capacity and delivery times.

3.3 Build Initial Loads

We first build loads by only enforcing vehicle capacity and maximum number of shipments per load. FTL shipments are loaded first followed by LTL and then Groupage. The initial loads will include all FTL and LTL shipments plus some Groupage ones. Delivery time violations occur in this stage but they are fixed in the next stage. The steps below are repeated for each cluster in order to build initial loads.

1. Estimate a lower bound V_{max} on the number of vehicles needed considering the total value of all shipments and the vehicle capacity.

2. Assign each FTL shipment to one new load, i.e. there will be X_{ftl} partially filled loads and $V_{max} - X_{ftl}$ 'empty loads' after this step.

3. Sort the X_{ftl} partially filled loads by increasing remaining volume.

4. Sort each of the X_{ltl} and X_{grp} shipment lists by decreasing size.

5. For each 'empty' load, assign the first shipment in X_{ltl} to the 'empty' load. After this, all the V_{max} are partially filled (i.e. with at least one FTL or LTL shipment assigned).

6. For each shipment s in the X_{ltl} list, find the loads with enough remaining capacity to take s. From those loads, select the one with an assigned shipment closest to s and then assign s to that load. If shipment s is part of a subgroup, check that vehicle capacity and maximum shipments per load are as specified. If there is no load with enough capacity for s, a new 'empty' load is created, i.e. V_{max} is increased by one. This step ends until list X_{ltl} is empty, i.e. until all LTL shipments have been assigned to loads.

7. The above step is then repeated for the X_{grp} list but no 'empty' loads are created. That is, some Groupage shipments might not be assigned to loads, instead they will be left for *parcel mode* transportation.

3.4 Carrier Assignment

In this stage, we have V_{max} loads that satisfy the vehicle capacity and that contain shipments that are *geographically compatible* which helps to identify the set of carriers that would be available (including number of vehicles available per carrier) to undertake each load. We use an Integer Programming (IP) assignment model to select a carrier for each load. The IP model seeks to minimise the overall transportation cost given by the assignment of each load to a particular carrier company. The cost obtained by the IP solver[2] is then used as an upper bound for the next stage of the approach that seeks to improve loads. The IP solver is executed again after each change to the plan during the improvement stage.

3.5 Improve Loads

In this stage, we have V_{max} loads each assigned to a particular carrier. The goal now is to move shipments between loads in order to improve the quality of the overall plan. This means mainly removing delivery time violations and reducing the overall cost. However, minimising total driving distance (and time) and maximising overall vehicle utilisation (fill factor) are also considered. Within a load, it is assumed that the first shipment is delivered at the earliest time, i.e. start of the delivery time window (or specific delivery time). At this point, the shipments in each load are sequenced from the nearest (to the source) to the farest and delivery time violations between consecutive shipments are identified.

[2]We employ the LP_Solve library available at http://lpsolve.sourceforge.net/5.5/

Move Shipments Between Loads in the Same Cluster. Using *swap* and *reassign* moves iteratively, LTL shipments in loads within the same cluster are moved in order to improve the quality of the current plan. Shipments are processed in descending order of the number of delivery time violations incurred. For a shipment, first a *reassign* move is tried and if the delivery time violation remains then a *swap* move is tried next. The number of delivery time violations is recalculated for each shipment before performing the next iteration. This *intra-cluster improvement* process is repeated until no delivery time violations remain or until no further improvement can be achieved.

Move Shipments Between Loads in Different Clusters. After the clustering stage, two shipments in different clusters might be separated by no more than 70 miles. This can occur because too large clusters are split into more clusters as explained in subsection 3.1. This process then tries to improve the quality of the plan by moving such shipments between different clusters. For each shipment in a cluster, the shipments in another cluster that are within 70 miles are put in a candidate list. Then the same *reassign* and *swap* moves are iteratively applied as before but now to shipments in different clusters. This *inter-cluster improvement* process is repeated until no delivery time violations remain or until no further improvement can be achieved.

Re-sequence Subgroup Shipments. Although no delivery time violations are considered between the shipments in a subgroup, sometimes after the above steps, another shipment in the same load might have a delivery time conflict with the subgroup. This is often solved by just re-sequencing the shipments in the subgroup. Moreover, re-sequencing the subgroup is always explored as this can improve cost or reduce backward mileage.

Re-sequence and Adjust Shipments in Loads. Exploring the re-sequencing of shipments within a load can help to eliminate the few delivery time violations that might still remain until this point. Next, adjustments are made to some loads in order to eliminate delivery time violations (if any) and further reduce cost. Two routines, applied to LTL and Groupage shipments, are used for this. One is to move shipments from loads to *parcel mode* transportation and the other is to reassign shipments between loads to reduce cost only.

4 Experiments and Results

We now describe the experiments and results used to evaluate the hybrid heuristic approach detailed above. We have applied our method to generate the transportation plans for several days using real-world data. We have compared these plans to those produced by the human planners at the company.

4.1 Problem Instances

We conducted experiments with 5 datasets each corresponding to a different planning day for one of 3TL's customers. Due to the commercial sensibility implications of revealing more details about the data used, at present we only provide a profile of the problem instances. An outline of each dataset is shown in Table 1 which shows for each problem instance: the total number of shipments to plan, the number of shipments of each type, the range of driving distances and range of driving times between the source and delivery points and also between delivery points. Since FTL shipments ($\geq 40\,m^3$) use at least half the capacity of a vehicle ($80\,m^3$), then the number of FTL in each problem instance indicates the minimum number of loads that will be formed. Groupage shipments usually make the largest proportion of shipments in the planning.

4.2 Results of Clustering and Subgrouping

Results of the clustering and subgrouping stages are shown in Table 2 where we can see the total number of clusters and subgroups identified by the algorithm. Since some of these clusters might contain only 1 point, we consider them as 'noise clusters'. Columns 3 and 4 of the table then show the number of 'no-noise clusters' and the range of points found in the set of clusters. For example, for dataset d107111 our adaptation of the DBScan algorithm identifies 13 clusters, none of them contains 1 point only. The

	Number of Shipments				Source to Delivery Points		Between Delivery Points	
	Total	FTL	LTL	Groupage	Distance (mi)	Time (min)	Distance (mi)	Time (min)
d170111	68	8	13	47	16.5 - 392.7	33 - 434	0 - 680.6	0 - 891
d180111	103	14	31	58	5.9 - 701.1	14 - 841	0 - 643.3	0 - 892
d190111	69	4	14	51	8.6 - 403.9	17 - 574	0 - 573.2	0 - 819
d200111	60	4	18	38	6.7 - 421.0	23 - 529	0 - 680.6	0 - 892
d210111	72	6	17	49	9.0 - 546.7	18 - 685	0 - 552.4	0 - 698

Table 1: Outline of the 5 problem instances used in the experiments. Columns 2-5 give the number of shipments to plan in each dataset. Column 6 (7) gives the range of driving distances in miles (driving times in minutes) between the source and delivery points. Column 8 (9) gives the range of driving distances in miles (driving times in minutes) between delivery points.

	Total Clusters	No-noise Clusters		Number of Subgroups	
	(with DBScan)	Number of Clusters	Points Per Cluster	2 Shipments	3 Shipments
d170111	13	13	2 - 13	3	4
d180111	16	14	2 - 16	5	10
d190111	13	10	3 - 12	2	3
d200111	11	9	2 - 12	2	4
d210111	13	11	2 - 9	1	9

Table 2: Results after the *clustering and subgrouping* stages. Column 2 gives the total number of clusters indentified by the modified DBScan algorithm. Column 3 gives the number of no-noise clusters, i.e. clusters containing at least 2 points. Column 4 gives the range of points per cluster. Columns 5 and 6 give the number of subgroups of size 2 and 3 identified in each dataset.

13 clusters contain a number of delivery points between 2 and 13. It should be noted that exactly the same delivery points are assigned to each cluster everytime the clustering stage runs on the same data. Columns 5 and 6 of the table show the number of subgroups identified. Remember that the shipments in a subgroup are then considered as a single shipment in the later stages of the algorithm. For example, Table 2 shows that for dataset d107111, out of the total 68 shipments, 18 of them are organised in 7 subgroups, 3 subgroups with 2 shipments each and 4 subgroups with 3 shipments each. For visualisation, Figure 1(a) shows the result of the clustering stage on the UK map for the dataset d107111.

4.3 Results of Building Loads and Assigning Carriers

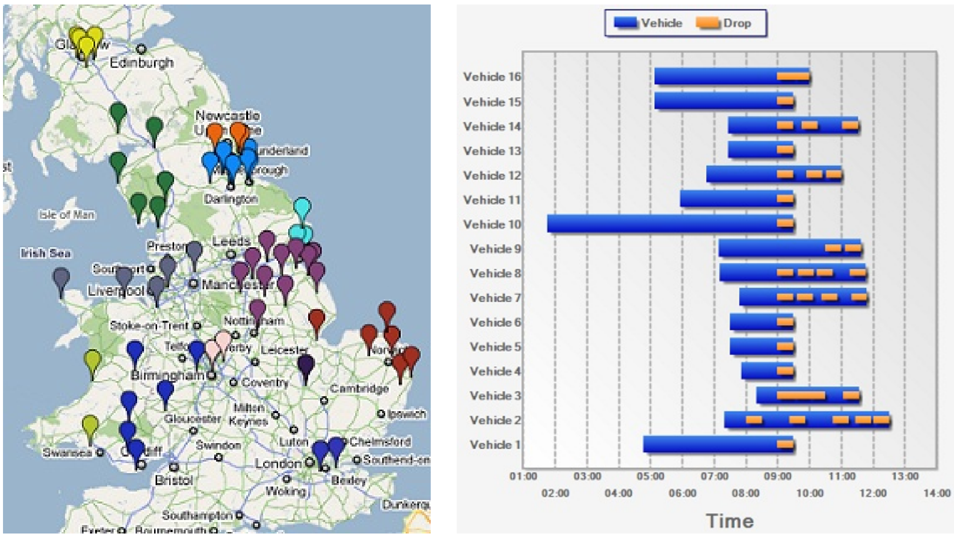
Results of the build initial loads and carrier assignment stages are shown in Table 3 where we can see the profile of vehicle loads generated by the hybrid heuristic. Each row in the table corresponds to the initial transportation plan for the given dataset. For example, for problem instance d107111, out of the 68 shipments, 38 are delivered in *load mode* using 16 loads and the other 30 shipments are delivered in *parcel mode*. On average each of the 16 loads fills 60.1% of the vehicle capacity. The delivery time of 2 shipments is violated in this initial plan. The total driving distance, driving time and cost (due to assigning loads to carriers) are also given in the table. We can see in Table 3 that these initial loads containg several delivery time violations, fixing them by moving shipments between loads while trying to the reduce overall cost (or minimise its increase) is the goal of the next stage in the algorithm.

4.4 Results of Improving Loads

Results of the loads improvement stage are shown in Table 4 where we can see the profile of vehicle loads after improvement from the initial loads in Table 3. Each row in Table 4 corresponds to the improved transportation plan for the given dataset. For example, for problem instance d107111, 3 shipments were moved from *load mode* to *parcel mode*. On average, the vehicle capacity fill factor for each of the 16

	Total Loads	Shipments Load/Parcel	Average Vehicle Fill Factor	Delivery Time Violations	Total Mileage	Total Driving Time (min)	Total Cost
d170111	16	38/30	60.1%	2	2241.8	2952	5023.7
d180111	26	73/30	74.2%	4	5690.9	7271	10436.2
d190111	13	38/31	59.8%	4	1808.2	2477	5377.6
d200111	14	42/18	57.2%	3	2675.7	3365	5323.7
d210111	15	41/31	61.7%	6	2896.5	4176	7095.2

Table 3: Results after the *build initial loads and carrier assignment* stages. Column 2 gives the total number of loads built while column 3 shows the split of shipments planned in load mode or in parcel mode. Column 4 gives the average percentage of filled capacity for those vehicles delivering the loads. Column 5 shows the number of shipments which delivery time is violated in these initial loads. Columns 6-8 show overall driving distance, driving time and cost for the delivery of loads.



(a) Example of clustering result (b) Example of plan result

Figure 1: Visualisation of transportation planning results for dataset d107111.

loads decreased slightly from 60.1% to 59.3%. There is 1 less delivery time violation in the improved plan. The driving distance and driving time were reduced but the cost increased slightly. We can see that this improve loads stage reduces to 0 or at most 1 the number of delivery time violations in all the datasets used in these experiments. In the cases where there is 1 delivery time violation, the vehicle arrives late to the delivery point by something between 2 and 3.5 hours. For visualisation, Figure 1(b) shows the resulting transportation plan for the dataset d107111. Each bar in the figure corresponds to a vehicle load. Each small bar inside a load bar represents a delivery, the longer inside bars (for loads 3 and 16) are delivery of a subgroup. That is, 30 of the 35 shipments delivered in *load mode* in this dataset d107111 are single-shipment deliveries. The other 2 deliveries are subgroups, 1 with 2 shipments and the other with 3 shipments. This also means that the number of subgroups was reduced from 7 (see Table 2) to 2 in this improve loads stage.

So far, 3TL has estimated that generating the transportation plans using the proposed hybrid algorithm instead of the current semi-automated approach, would generate a cost saving of between 500 and 1500 GBP per day corresponding to a saving of around 8% in transportation costs. For the 5 days planning used in the experiments of this paper, the actual overall cost of the transportation plans created by the human planners is 59874 GPB. The estimated cost of the improved plans generated automatically with our algorithm is 34280 GPB which represents a saving of around 25594 GPB or 42.7% for the week. However, we should mention that this is the maximum saving that could be achieved but the real cost can only be calculated in the day of operations. When estimating the cost for the improved plans, we

	Total Loads	Shipments Load/Parcel	Average Vehicle Fill Factor	Delivery Time Violations	Total Mileage	Total Driving Time (min)	Total Cost
d170111	16	35/33	59.3%	1	2024.4	2717	5155.7
d180111	26	71/32	74.0%	0	5792.3	7442	10525.4
d190111	13	35/34	58.9%	1	1788.6	2226	5539.3
d200111	14	41/19	57.0%	1	2692.1	3476	5650.8
d210111	15	34/38	57.8%	0	2895.1	4124	7409.5

Table 4: Results after the *improve loads* stage. Column 2 gives the total number of loads built while column 3 shows the split of shipments planned in load mode or in groupage mode. Column 4 gives the average percentage of filled capacity for those vehicles delivering the loads. Column 5 shows the number of shipments which delivery time is violated in these improved loads. Columns 6-8 show overall mileage, driving time and cost for the delivery of loads.

assume that the carrier company representing the best assignment to each load is available. This might not be the case on the day of operations because the selected carrier company might not have vehicles available so the next best available carrier (perhaps more expensive) will be selected to deliver the load. The plans produced so far seem to be practical and also better than those produced by the experienced human planners at 3TL.

5 Final Remarks

This paper presented a hybrid heuristic approach to tackle a transportation planning problem arising at a UK-based company that provides logistic services. The problem tackled involves assigning shipments to loads, routing the delivery of shipments in each load and assigning the loads to a number of different carrier companies. The hybrid heuristic approach incorporates a clustering algorithm, a constructive heuristic, a local search phase and an integer programming assignment procedure. The proposed method has been tested with real-world data and results show that the transportation plans generated are practical and better compared to those produced at present by the expert human planners. The company is currently in the process of deploying the hybrid heuristic within their bespoke multi-mode transportation management system.

References

[1] Patricia J. Daugherty, Alexander E. Ellinger, and Craig M. Gustin. Integrated logistics: achieving logistics performance improvements. *Supply Chain Management: An International Journal*, 1(3):25–33, 1996.

[2] Samuel Eilon and Nicos Christofides. The loading problem. *Management science*, 17(5):259–268, 1971.

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD 1996)*, pages 226–231. AAAI Press, 1996.

[4] John Mangan, Chandra Lalwani, and Tim Butcher. *Global logistics and supply chain management*. Wiley, 2008.

[5] D. Simchi-Levi, X. Chen, and J. Bramel. *The logic of logistics: theory, algorithms and applications for logistics management*. Springer, 2005.

[6] Marius M. Solomon. Algorithms for the vehicle routing problem and scheduling problems with time window constraints. *Operations research*, 35:254–265, 1987.