Metaheuristics for the Pick-Up and Delivery Problem with Contracted Orders

Philip Mourdjis[1], Peter Cowling[1], and Martin Robinson[2]

[1] York Center for Complex Systems Analysis (YCCSA) and Department of Computer Science, University of York, York, UK
{pjm515,peter.cowling}@york.ac.uk
[2] Transfaction Ltd., Suffolk, UK
martin.robinson@transfaction.com

Abstract. Contracted orders represent a novel extension to the Pick-up and Delivery Problem (PDP) with soft time windows. This extension to the multiple depot problem has depots managed by separate, competing haulage companies "carriers". Orders may be assigned to a specific carrier "contracted", "allocated" to a specific carrier but allowed to swap if this improves the solution or free to use any carrier "spot hired". Soft time windows lead to a multi-objective problem of minimising distance travelled and delay incurred. In this paper we use real order data supplied by 3 large distributors and 220 carriers. Additional, randomised, orders are generated to match the distributions observed in this data, representing backhaul orders for which no data is available. We compare a manual scheduling technique based on discussions with industry partners to popular metaheuristics for similar problems namely Tabu Search (TS), Variable Neighbourhood Search (VNS) and Hybrid Variable Neighbourhood Tabu Search (HVNTS), using our modified local search operators. Results show that VNS and HVNTS produce results which are 50% shorter than greedy approaches across test instances of 300 orders in a one week period.

1 Introduction

The purpose of this paper is to compare the effectiveness of a number of heuristic methods on a specific real world Vehicle Routing Problem (VRP), specifically a multi-depot VRP with pick up and delivery and soft time windows. Our research focusses on medium to long distance deliveries made from point to point within the UK. By considering sample orders from 3 large distributors and 220 haulage companies "carriers" we aim to reduce transportation costs and carbon emissions through the intelligent coordination of logistics activities. We are particularly interested in the gains possible through the re-assignment of orders between carriers. We note that there are currently 3 ways an order can be specified to carriers:

Contracted orders must be serviced by a specified carrier, the order may be re-allocated only between trucks belonging to this carrier.

C. Blum and G. Ochoa (Eds.): EvoCOP 2014, LNCS 8600, pp. 170–181, 2014.

Allocated orders are assigned to specific carriers but may be re-allocated by that carrier to sub-contracted carriers.

Spot hired orders may be assigned to any truck belonging to any carrier and re-assigned any number of times to any truck.

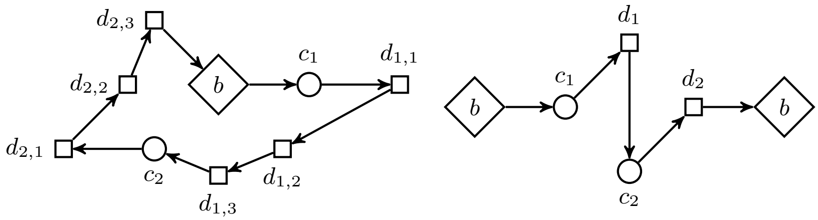
The remainder of this paper is organised as follows. Section 2 introduces the ideas, models and concepts that we build upon in this paper. Section 3 sets out our model. Section 4 presents the local search operators used including our modification of GENI and describes the various metaheuristic methods chosen for experimentation along with the changes made to them to fit our model. Section 5 details the parameters used in our experiments, how randomised orders were generated from our existing data and compares the effectiveness of the introduced metaheuristics with varying levels of contracted and allocated orders. Section 6 presents conclusions and results analysis. Finally, we present an outline of areas for future research work.

2 Related Work

The work we have undertaken builds on VRPs with Time Windows (VRPTW), where orders must be fulfilled between a given earliest and latest time. These problems are summarised with an overview of exact algorithms and optimisation methods by Desrosiers et al. [1] and more recently with local search algorithms and metaheuristic approaches by Bräysy and Gendreau [2,3]. We also build upon research into Pick-up and Delivery Problems (PDPs) recently classified and summarised by Berbeglia et al. [4] and Parragh et al. [5]. The combination of these two areas is the PDP with Time Windows (PDPTW) [6] and still represents a lesser researched area than either of its parents. Ropke et al. [7] proposes an exact solution for the PDPTW while Malca and Semet [8] present a Tabu Search (TS) approach. Gendreau et al. [9] present neighbourhood searches for the dynamic version of this problem. For our real world problem we considered local search neighbourhoods and metaheuristics that have proved strong in the related VRPTW and tailor their methods to our specific needs. Taillard et al. [10] and Cordeau et al. [11,12] present techniques for implementing TS algorithms similar to those we use in this paper. Variable Neighbourhood Search (VNS) originally introduced by Mladenovic and Hansen [13] is a very good, general purpose, search metaheuristic capable of adapting to a wide variety of applications [14]. VNS has since been successfully applied to the VRPTW by Bräysy [15] and the multi-depot VRPTW by Polecek et al. [16]. The recent Hybrid Variable Neighbourhood Tabu Search (HVNTS) method of Belhaiza et al. [17] is tailored to the VRP with multiple time windows and was found to compare favourably to an ant colony optimisation on instances of the problem studied there.

3 Problem Definition

The PDP with contracted orders is defined on a directed graph $G = (V, A)$ where A is the arc set and $V = \{B, N\}$ is the vertex set split into B base-depot locations and N customer locations. A carrier is defined as a base location $b_i \in B$ and a set of trucks $T_i = \{T_i^0, \ldots, T_i^{M_i}\}$ where M_i is the number of trucks for carrier i. An order i consists of a collection location $c_i \in N$ and a final delivery location $d_i \in N$. In reality there are often several delivery locations as shown in Fig. 1a but at present we treat these orders as atomic, with the complexity of additional delivery locations abstracted away as in Fig. 1b for simplicity. The problem involves routing n orders into m routes, allowing for zero cost empty routes. Minimising m is not considered as part of this problem though it is kept low as a side effect of the heuristics used.



(a) Real (b) Representation

Fig. 1. Route abstraction

Distance Model. We denote $c_{i,j}$ as the cost of travelling from order i to order j, due to the route abstraction shown in Fig. 1 these costs are asymmetric such that $c_{i,j} \neq c_{j,i}$. the straight line distance travelled while empty between the last delivery of the previous order and the current orders collection is used as the cost (Thus if a truck has no orders it has no associated cost). Each route is terminated at both ends by a dummy order located at the specified trucks base depot, thus a route j with k orders has dummy orders at 0 and $k+1$, its empty distance cost, d_j, is shown in equation 1, constrained by Max_D, the maximum distance a truck is permitted to drive in a week.

$$d_j = \sum_{i=0}^{k} c_{i,i+1}.\text{where } d_j < \text{Max}_D \tag{1}$$

Any change to the solution can be mapped to a series of insertion and removal operations. As the orders themselves are present in both solutions (before and after any change) the only aspects that need to be considered are the legs between orders, shown in Fig. 2. Denoting $\check{c}_i$ as the insertion cost of an order i between two pre-existing orders x and y, the change caused by inserting an order is calculated as shown in equation 2.

$$\check{c}_i = c_{x,i} + c_{i,y} - c_{x,y}. \tag{2}$$

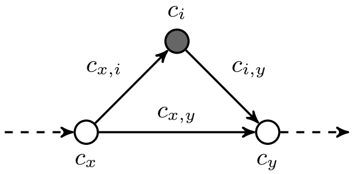


Fig. 2. Route alteration

Similarly the removal cost $\hat{c}_i$ of an order is as shown in equation 3:

$$\hat{c}_i = c_{x,y} - c_{x,i} - c_{i,y}. \tag{3}$$

A key point to note is that both $\hat{c}_i$ and $\breve{c}_i$ may be positive or negative with positive costs indicating an increase in empty distance and negative costs indicating a decrease.

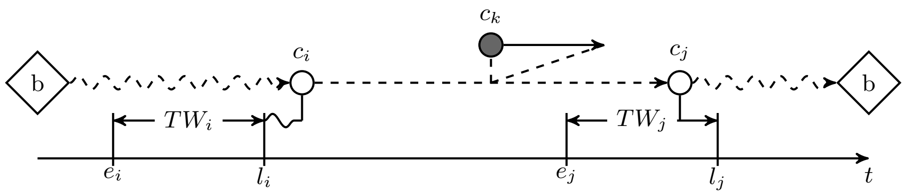
3.1 Time Window Model

Figure 3 shows a number of collection time windows. e_i is the earliest time a truck may service customer c_i and l_i is the latest time, there is no penalty for arriving at a location early, though the truck will have to wait until the specified earliest time to be serviced. If the truck arrives after l_i the order is said to be delayed by $t_i - l_i$ where t_i is the actual time customer i is serviced. Not shown in Fig. 3 is the service time required for loading / unloading at a customer location, this is denoted by s_i. Tardiness is calculated based on the vehicles arrival time at a location thus if a vehicle arrives at the latest arrival time the tardiness is 0 even though the truck will not leave until $s_i + l_i$ (after the latest time l_i). The tardiness of a vehicle t_V is simply the sum of the individual delays experienced at each location in its route. Orders are always inserted as early as possible at the chosen insertion point and changes to a route force an update of delay parameters for each subsequent location.

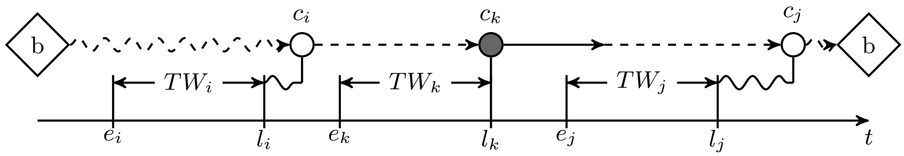
Note, that since a collection node must occur before its delivery node, reversing a section of a route will significantly alter the distance, time windows are also usually tight enough such that one or more orders will be rendered significantly delayed. Methods relying on partial route inversions such as GENI [18] and iCROSS [15] will therefore not work well without alteration.

3.2 Objective

Our optimisation procedure seeks to fulfil all orders in such a way as to reduce the total travel cost whilst keeping tardiness to a minimum. The fitness of a vehicles route, f_V, is given in equation 4. Here α represents a tunable parameter between 0 and 1 determining the relative importance of tardiness and distance respectively. D_V and T_V represent the total distance and time of a vehicles route



(a) Before



(b) After

Fig. 3. Tardiness - Two orders c_i and c_j are currently scheduled and only c_i is delayed. If another order (c_k) were placed between the two existing orders, c_i would remain at its current time while c_j may have to occur later, potentially becoming delayed.

respectively, dividing by these gives relative empty miles and relative tardiness, allowing comparisons to be made between the two metrics which would otherwise be orders of magnitude different.

$$f_V = \alpha \left(\frac{d_V}{D_V} \right) + (1 - \alpha) \left(\frac{t_V}{T_V} \right) \tag{4}$$

Since the impact of both of these upon a carrier is in additional cost (or lost profits) we combine them into a single objective function. α therefore determines the relative cost of driving additional miles versus late delivery penalties. The fitness of a solution is the sum of the individual fitnesses of all its vehicles as shown in equation 5.

$$F_S = \sum_{j=1}^{m} f_j. \tag{5}$$

4 Solution Methods

4.1 Local Search Operators

For hundreds of carriers and thousands of orders it is computationally intensive to calculate a fitness from a solution. We use local moves to make incremental changes to the current solution instead and measure changes in fitness. These are much easier to compute and over successive iterations we can make large changes to the solution.

A number of local search operators are used including cross [10], relocate [2], swap [2] and a modification of GENI [18]. Each of these local moves is intended

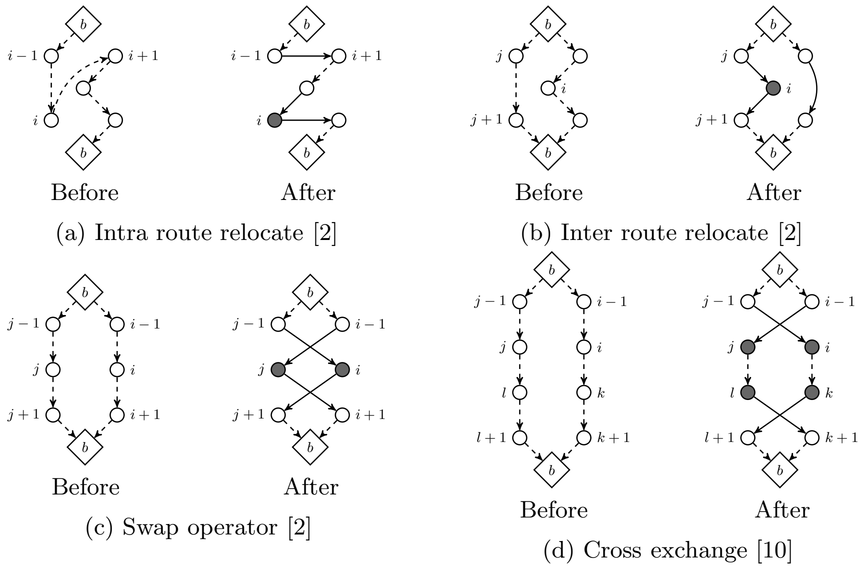


Fig. 4. Local moves

to preserve existing orderings as much as possible. Our modification of GENI is presented below and the other operators used are summarised in Fig. 4. They represent restricted 3- and 4-opt operators which preserve the order of nodes.

GENI - Preserve Ordering. A local move, similar in spirit to the generalized insertion (GENI) procedure of Gendreau et al. [18] was devised as follows, for a given order to be inserted into a chosen target route, for each pair of nodes in the target route, calculate the two insertion costs as shown in Fig. 5b using equation 2. In comparison to GENI, Fig. 5a, GENI-PO does not reverse the traversal of any existing arcs of the solution and should be more suitable for this real world problem with time windows.

4.2 Metaheuristics

We sought to make comparisons between popular and contemporary metaheuristics from the literature and a greedy assignment without optimisation. We use simple versions of TS [19,10], VNS [13,14] and HVNTS [17] for our experiments, adapted such that they are effective for our problem and a fair comparison can be made between them. To this end a number of differences to the original methods have been made.

1. We have adapted each procedure to use the same set of local search operators, namely intra- and inter-route relocate, swap, cross and GENI-PO introduced in Section 4.1. When optimising routes, we check if the order we want to move is contracted, if it is, we can still move the order between vehicles but

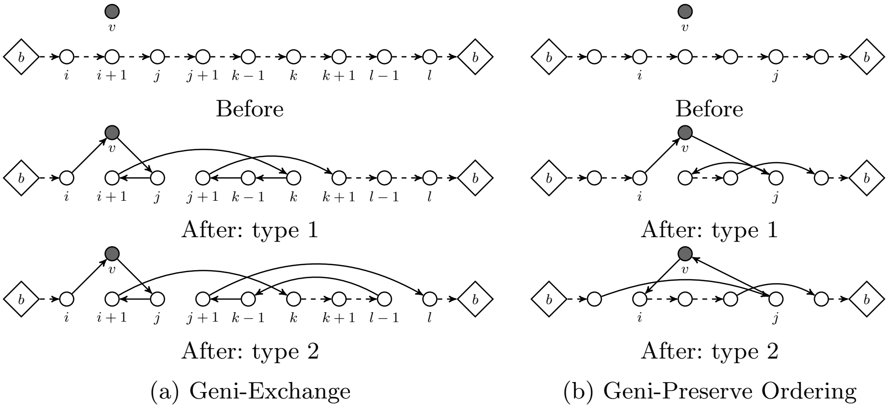


Fig. 5. GENI and GENI-PO local moves

the search space becomes restricted to only those vehicles belonging to the contracted carrier.

2. In each case a greedy insertion method is used to generate an initial solution. This method takes a random list of orders and inserts each in its lowest cost insertion location given already scheduled orders. For each insertion location, the lowest insertion cost $min\left(\check{c}_i\right)$ is determined by using equation 2 for each valid insertion point. The greedy insertion method is used as a baseline for the comparison after discussions with our sponsor, Transfaction, as a technique which closely mimics current manual/semi-automatic approaches to scheduling.
3. Once an initial solution has been generated the three methods are each given 50 CPU seconds to generate a result, allowing fast iterating techniques to run more iterations. As the heuristics use differing amounts of CPU time to process one iteration comparing the heuristics with a fixed number of iterations would not be fair. In the extreme a heuristic approach may be "beaten" by an exhaustive style search if the same iterations of each were performed but would take substantially (thousands of times) longer to run. We feel that keeping CPU time constant is a fair way to evaluate these methods [20]. All tests were carried out on a Windows 7 SP1 desktop machine running C$^\sharp$ code on a 2.8Ghz Intel core i5-2300 processor with 8Gb of RAM.

5 Computational Experiments

5.1 Generating Orders

At present we have access to one week's worth of order data for 3 large distributors and 220 hauliers. To more thoroughly test our heuristics and to ensure we are not overfitting to our sample data we generate additional randomised orders in the form of backhauls [4]. These are derived from existing orders where collection locations are picked from real delivery locations and delivery locations are picked from real collection locations[1]. We use a set of uniform and Poisson distributions with parameters tuned to approximate the orders for which we have data.

5.2 Speed and Travel Parameters

Each collection and delivery location used is based on UK postcodes which are translated to standard eastings and northings. Between locations, straight line distances are used and it is assumed that trucks travel at a constant 35 Kph as this is the average value derived from our massive data set. We set Max_D at 1650 km as at 35 Kph this is the limit for the number of hours a long haul driver is allowed to drive in a week.

5.3 Aims

To investigate the effect of contracted and allocated orders, we conducted 5 sets of experiments, in each set, orders were defined with: 100% contracted; 30% contracted, 60% allocated & 10% spot hired; 30% contracted & 70% spot hired; 60% allocated & 40% spot hired and all spot hired respectively. In each case 10 seeded randomised runs were performed for each heuristic. For each run, 200 real orders were selected from a database of orders and a further 100 were generated as described in Section 5.1.

5.4 Findings

To easily display the large numbers of results generated, groups of four box plots have been used to represent the four techniques compared. Each box plot represents the min / max and quartiles of the 10 runs. Figure 6 presents this information along with the order of the heuristics used in the following charts. Here "Greedy" represents the initial solution before optimisation, VNS, TS and HVNTS represent the results from our modified heuristics after optimisation. Figures 7 and 8 present the empty miles and delay for all heuristics across the range of scenarios introduced above. We can see that there is a clear trend towards shorter distances as we allow orders more freedom in carrier choice. This trend is amplified by our metaheuristics which produce little benefit in a fully contracted model but produce benefits of approximately 50% in the spot model.

Of the three heuristic approaches investigated, VNS and HVNTS can be seen to produce the shortest routes across multiple runs of our experiments when orders are spot hired. When all orders are contracted to a specific carrier there is much less variation in the results observed, note that there is still a large

[1] We know backhaul orders of this kind exist but have no data for them, since they are often requested by small distributors.

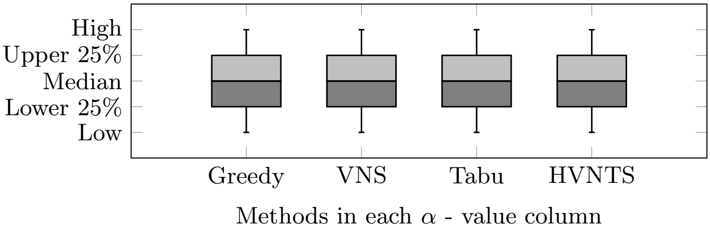


Fig. 6. Key

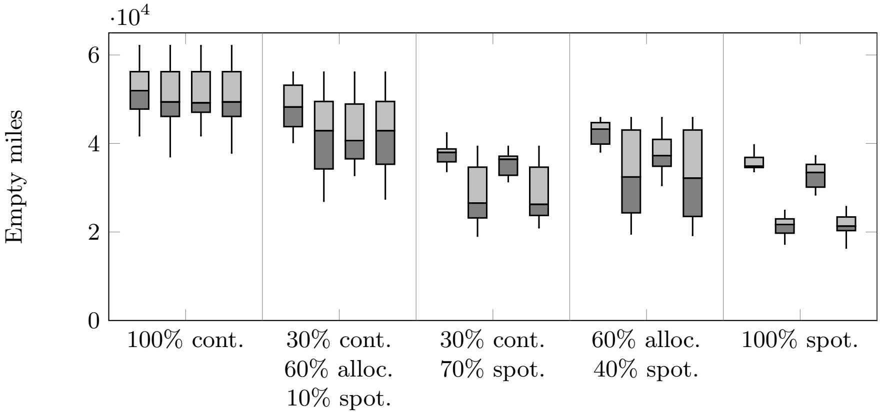


Fig. 7. Empty miles - lower is better

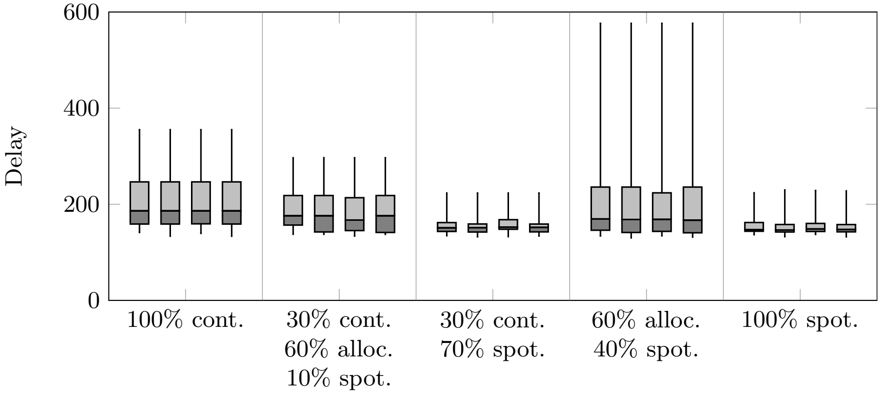


Fig. 8. Delay - lower is better

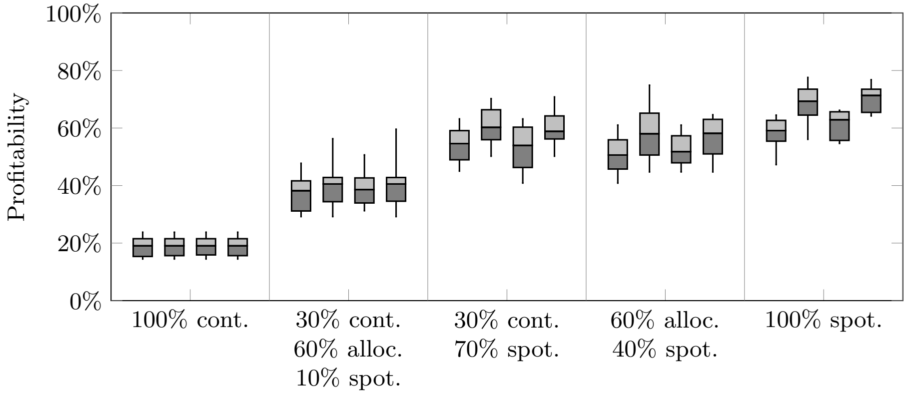


Fig. 9. Average carrier profitability - higher is better

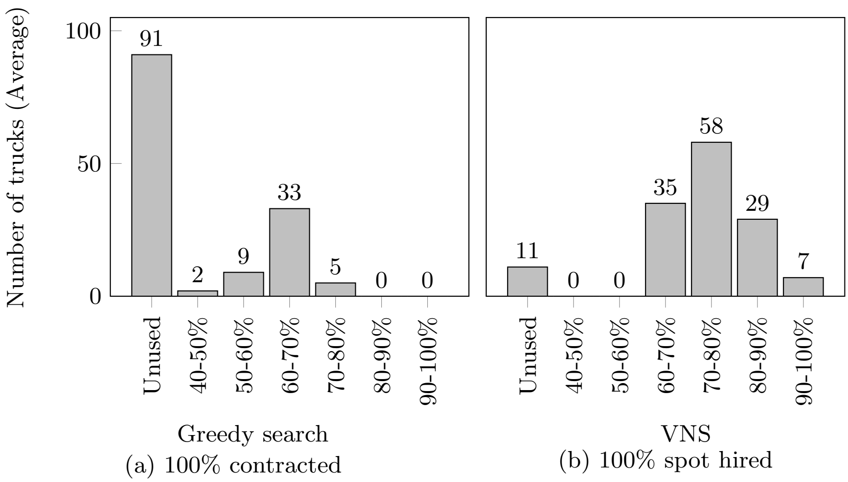


Fig. 10. Route profitability breakdown

solution set to evaluate in this case as even when contracted to a carrier there remains a choice of delivery vehicle and ordering. We feel that due to time order constraints the greedy insertion heuristic used in the entirely contracted examples is able to produce routes that are close to optimal and do not leave room for our heuristic techniques to improve upon.

Figure 9 shows that the average profit a carrier attains under any heuristic increases as the proportion of spot hired orders increases. Here profitability is the percentage of distance travelled that is spent on delivery, between pick-up and delivery.

We also observed that the fully contracted scenario using greedy scheduling produced an unfair distribution of orders between carriers such that many were left without any orders (Fig. 10a). Moving to the other end of the spectrum, VNS on the fully spot hired scenario produces higher rates of profitability which are consistent across many more carriers (Fig. 10b), yielding a more sustainable situation from the point of view of the carriers.

6 Conclusions and Future Work

Conclusions. From our results it is clear to see that the shortest routes are achieved when all orders are spot hired, free to be assigned to any carrier, and that these routes offer no significant change in the overall delay of the solution. We note that pre-allocating orders to preferred carriers, though better than being contracted, still produces relatively long routes with the optimisation procedures we have used here. Also of interest is that our optimisation strategies produce far greater benefits over the initial solution when there are no contracted or allocated orders. This can likely be attributed to the larger solution space available to explore.

We believe that since spot hired orders are more efficient for carriers to handle, they can be delivered more cheaply. A coordinating body such as our industrial partner, Transfaction, therefore has the potential to deliver on its promises of increasing carrier profits, reducing distributor costs and reducing carbon emissions in the delivery chain.

As we increase the number of orders from 300, we expect the efficiency gains from allowing more freedom in scheduling (through more spot orders) and through the use of effective heuristics will be even higher.

Future Work. We aim to adapt the current model and techniques of our problem to a dynamic environment where orders arrive in real time, in this case the most suitable heuristic may change and further alterations to our model, local move operators and metaheuristic approaches may be needed. Also the heuristics and local search operators need to be significantly sped up. Further research intends to investigate the effects of combining orders where possible, for example, truck capacity may allow more than one order to be collected before deliveries commence.

Acknowledgements. This work has been funded by the Large Scale Complex IT Systems (LSCITS) EngD EPSRC initiative in the Department of Computer Science at the University of York and by Transfaction Ltd.

References

1. Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F.: Time Constrained Routing and Scheduling. Handbooks in Operations Research and Management Science 8, 35–139 (1995)
2. Bräysy, O., Gendreau, M.: Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. Transportation Science 39(1), 104–118 (2005)

3. Bräysy, O., Gendreau, M.: Vehicle Routing Problem with Time Windows, Part II: Metaheuristics. Transportation Science 39(1), 119–139 (2005)
4. Berbeglia, G., Cordeau, J.F., Gribkovskaia, I., Laporte, G.: Static pickup and delivery problems: a classification scheme and survey. TOP 15(1), 1–31 (2007)
5. Parragh, S.N., Doerner, K.F., Hartl, R.F.: A survey on pickup and delivery problems. Journal für Betriebswirtschaft 58(2), 81–117 (2008)
6. Dumas, Y., Desrosiers, J., Soumis, F.: The pickup and delivery problem with time windows. European Journal of Operational Research 54, 7–22 (1991)
7. Ropke, S., Pisinger, D.: An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. Transportation Science 40(4), 455–472 (2005)
8. Malca, F., Semet, F.: A tabu search heuristic for the pickup and delivery problem with time windows and a fixed size fleet, 1–5 (2003) (unpublished manuscript)
9. Gendreau, M., Guertin, F., Potvin, J.Y., Séguin, R.: Neighborhood Search Heuristics for a Dynamic Vehicle Dispatching Problem with Pick-ups and Deliveries. Transportation Research Part C: Emerging Technologies 14(3), 157–174 (2006)
10. Taillard, E.D., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.Y.: A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. Transportation Science 31(2), 170–186 (1997)
11. Cordeau, J.F., Laporte, G., Mercier, A.: A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society 52(8), 928–936 (2001)
12. Cordeau, J.F., Laporte, G., Mercier, A.: Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. Journal of the Operational Research Society 55(5), 542–546 (2004)
13. Mladenović, N., Hansen, P.: Variable Neighbourhood Search. Computers & Operations Research 24(1), 1097–1100 (1997)
14. Hansen, P., Mladenović, N., Moreno Pérez, J.A.: Variable Neighbourhood Search: Methods and Applications. Annals of Operations Research 175(1), 367–407 (2009)
15. Bräysy, O.: A Reactive Variable Neighborhood Search for the Vehicle-Routing Problem with Time Windows. INFORMS Journal On Computing 15(4), 347–368 (2003)
16. Polacek, M., Hartl, R.F., Doerner, K.F.: A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with Time Windows. Journal of Heuristics 10, 613–627 (2004)
17. Belhaiza, S., Hansen, P., Laporte, G.: A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. Computers & Operations Research (August 2013)
18. Gendreau, M., Hertz, A., Laporte, G.: New Insertion and Post Optimization Procedures for the Traveling Salesman Problem. Operations Research 40(6), 1086–1095 (1992)
19. Glover, F.: Artificial Intelligence, Heuristic Frameworks and Tabu Search. Managerial and Decision Economics 11(5), 365–375 (1990)
20. Barr, R.S., Golden, B.L., Kelly, J.P., Resende, M.G.C., Stewart, W.R.: Designing and Reporting on Computational Experiments with Heuristic Methods. Journal of Heuristics 1(1), 9–32 (1995)