The Effect of Cooperation in Pickup and Multiple Delivery Problems

Philip Mourdjis[1], Fiona Polack[1], Peter Cowling[1], Yujie Chen[1] and Martin Robinson[2]
[1]*YCCSA, University of York, York, U.K.*
[2]*Transfaction Ltd., Cambridge, U.K.*

Keywords: Vehicle routing, Pickup-and-delivery, Real-world Problem.

Abstract: Small logistics companies operate in many towns and cities across the UK, and need to be able to compete with larger delivery companies who can leverage economies of scale to provide lower costs to customers. If small companies were willing to work together, all could benefit from reduced operating costs, enabling them to compete and survive against larger delivery companies. In cooperation with Transfaction Ltd., we investigate dynamic scheduling of shared loads for real-world, long distance truck haulage in the UK. We model the problem as a dynamic pickup and multiple delivery problem (PMDP). The PMDP is a one-many problem (one pickup, many drop-offs), unlike the more widely researched one-one (pickup and delivery problem, PDP) and one-many-one (vehicle routing problem, VRP) problems.

1 INTRODUCTION

With over six thousand hauliers in the UK alone (Dff International Ltd, 2015), competition is fierce. Hauliers face the orthogonal demands of short notice from customers, an expectation of low-cost service, and environmental sustainability concerns (McLeod et al., 2012; Nahum, 2013; Demir et al., 2014). Because larger carriers can leverage economies of scale to benefit in routing and scheduling, competition is getting ever stronger. If smaller carriers could work together, they could increase scheduling efficiency, save on mileage costs, and improve flexibility. In this paper we quantify the savings possible when carriers outsource some of their customer consignments to other carriers, working independently or as a group.

As a real-world problem, there are constraints that must be satisfied, such as vehicle capacity, soft time windows and driver working hour rules. The problem is defined in terms of consignments which include a single pickup location and one or more delivery locations. Consignments vary in size, and may be able to share one delivery vehicle, to save cost. A key constraint is that each vehicle must be unloaded in the reverse order to the loading order: deliveries from one vehicle are constrained to a last-in, first-out (LIFO) order. Concretely, consignment A may be interrupted by another if all of the second consignment's deliveries are serviced before continuing with consignment A's deliveries. We call this a *pickup and multiple delivery problem* (PMDP). This paper investigates the cost savings which are possible if carriers distributed across a country share consignments.

2 RELATED WORK

Research on PDPs usually concentrates on static models of small scale problems such as servicing taxi requests, or ride sharing schemes (Toth and Vigo, 1997) – dial-a-ride problems (DARPs). Desaulniers et al. (2002) present a widely accepted mathematical formulation for the generic PDP, which they refer to as the *vehicle routing problem with pickup and delivery and time windows*.

Variations of the PDP handle constraints on the number of vehicles used, time windows on requests, capacities and number of depots. However, most of the existing research is on static problems, in which all requests are known in advance (Berbeglia et al., 2007). Exact solutions to static PDPs favour branch-and-cut-and-price algorithms using column generation techniques, for example, Dumas et al. (1991) uses this approach to solve a multi-depot PDP for problems with up to 55 requests. No indication is given of whether their approach scales to larger problem sizes.

Exact solutions to dynamic problems include a variation of the column generation approach (Gschwind et al., 2012), used to solve DARPs of up to 96 requests, with either static or dynamic time win-

Mourdjis, P., Polack, F., Cowling, P., Chen, Y. and Robinson, M.
The Effect of Cooperation in Pickup and Multiple Delivery Problems.
DOI: 10.5220/0005748902870295
In *Proceedings of 5th the International Conference on Operations Research and Enterprise Systems (ICORES 2016)*, pages 287-295
ISBN: 978-989-758-171-7

dows. Xu et al. (2003) solve a PDP based on real-world logistics with multiple carriers, vehicle types and LIFO constraints using a set partitioning formulation containing an exponential number of columns. However, in general, exact methods do not scale well, so heuristic and hyper-heuristic approaches that can quickly find near-optimal solutions, have become popular for large-scale, real-world problems. Laporte (2009) provides a good overview of exact and heuristic methods for vehicle routing problems. More recently, heuristic approaches have been applied to scheduling with LIFO loading constraints (Cherkesly et al., 2015; Crainic et al., 2015; Benavent et al., 2015).

Cherkesly et al. (2015) use a three phase approach. First, multiple routes are created using a greedy randomised adaptive search procedure; next variable neighbourhood descent (VND) applies local search to derive new solutions using a diversification strategy derived from Rochat and Taillard (1995). Finally, crossover is used to combine solutions to form further candidate solutions. Benavent et al. (2015) use a multi-start tabu search approach that uses Clarke and Wright savings (Clarke and Wright, 1964) as well as two random schedule heuristics to build seed routes. The tabu search improves solutions by repeatedly removing and re-inserting consignments, using traditional strategies to prevent cycling and promote diversification.

Existing approaches to dynamic scheduling of PDPs (summarised in Bräysy and Gendreau (2005b)) often use a two-phase hyper-heuristic (Berbeglia et al., 2010): requests are first inserted into a schedule, then optimisation is performed, either on a route that has been changed or on an entire schedule. Research has focused on different insertion, removal and local search operators, and on the heuristics that choose between operators at any point. For example, Gendreau et al. (2006) use neighbourhood search heuristics and ejection chains to tackle same-day courier PDP. Mitrović-Minić et al. (2004) use a double horizon approach with routing and scheduling sub-problems to schedule similar problems of a larger size. Albareda-Sambola et al. (2014) use probabilistic information to inform their routing of a multi-period VRP.

We are concerned with efficient solution of scheduling under just-in-time logistics, where the customer expectation is that hauliers respond quickly to delivery requests, and where same-day delivery often attracts premium payment rates. In the traditional approach used by small haulage companies, static scheduling is re-run daily. However, static scheduling cannot be used for real-time response to orders, and does not take account of the existing schedule and loading. We propose a dynamic scheduler that intelligently adapts to incoming requests, a novel variant of dynamic PDP (Berbeglia et al., 2010).

Our model of the PMDP is based on the generic PDP model of Desaulniers et al. (2002). Our variable neighbourhood descent with memory scheduling (VNDM) takes inspiration from the hybrid variable neighbourhood tabu search (VNTS, Belhaiza et al. (2013)), which outperforms tabu and variable neighbourhood approaches for static VRPs. A schedule is built up by repeatedly inserting requests then performing optimisation. The strict LIFO constraint in PMDP, along with constraints such as the vehicle capacity, makes it difficult to find improving moves in PMDP, so we develop a descent based algorithm and local search operators tailored to PMDP, with roots in classic VRP and PDP solutions. Once a solution has been built, we perform optimisation whilst aiming to minimise ordering inversions within a vehicle's schedule, as these are unlikely to improve results in problems with tight time windows and LIFO constraints on deliveries. Local search techniques that affect delivery order, such as those presented by Taillard et al. (1997) and Bräysy and Gendreau (2005a), and the GENI technique (Gendreau et al., 1992), are unsuitable for direct use on our problem because they cause large changes in schedule ordering.

3 MODEL

The PMDP is defined on a directed graph $DG = (N,A)$ where A is the arc set and N is the node set. Each request r is identified by (n_r, l_r t_r^{start}, t_r^{end}, $tt_r^{service}$) where n_r is the location, l_r is the load (where the summation of pickup load and delivery loads for a consignment is equal to zero). t_r^{start}, t_r^{end}, represent the start and end time of the arrival window respectively where the service time $tt_r^{service}$ must begin (for clarity we use double letters to represent quantities). R is the set of requests where $R = P \cup D \cup O$, P being the set of pickup-requests and D the set of delivery-requests. O is the set of origins which are dummy requests used to represent the multiple depot locations of the problem. The arc between two requests r and u (that is, between nodes (n_r, n_u)) is the arc (r,u). A consignment c is identified by (p_c, D_c, t_c) where p_c is the pickup-request and $D_c = d_c^1, \ldots, d_c^{nc}$ is the sequence of delivery-requests. Each consignment has a received time t_c, which is the time at which the order is entered in the system. C is the set of consignments. $A_k \subset A$ represents the feasible arcs for vehicle k. The binary flow variable b_{ruk} is set to one if arc $(r,u) \in A_k$ is used by the vehicle k, and to zero otherwise. ll_{rk}

is the load of vehicle k at request r and is not fixed but dependent on the other arcs in the vehicle's route. It is calculated as a running sum where each request either adds to the load (pickup) or subtracts from the load (delivery). A vehicle starts and ends its route at one of the depots with load equal to zero.

The goal is to minimise the total cost of servicing all requests $r \in R$:

$$\min \sum_{k \in K} \sum_{(r,u) \in A_k} C_{ruk} * b_{ruk} \tag{1}$$

where:

$$C_{ruk} = nc(nn_{ru}, ll_{rk}) + tc(ruk) + dc(tt_{uk}^{delay}) \tag{2}$$

subject to the constraints in the appendix. C_{ruk} is the cost of vehicle k servicing (r,u), calculated using running cost estimations for a 44-tonne articulated truck in 2014 based on data from the UK Road Haulage Association (RHA) (Dff International Ltd, 2014). The component costs are: $nc(nn_{ru}, ll_{rk})$, the cost of travelling distance nn_{ru} (the length of arc (r,u)) with load ll_{rk}; $tc(ruk)$, the cost of the time taken by vehicle k to travel arc (r,u), and $dc(tt_{rk}^{delay})$, the cost of the penalty for arriving at request u late, we use a stepwise function (increasing every hour) after an initial grace period, in line with industry practice. Consignments may be either customer orders or backhauls (post-delivery return to pickup location, for instance to dispose of packaging), these differ only in that backhauls are usually mostly empty loads.

4 SOLUTION APPROACH

Our PMDP solution approach has three components: discrete event simulation (DES), the variable neighbourhood descent with memory (VNDM) hyper-heuristic, and the low level heuristics (LLHs).

4.1 Discrete Event Simulation (DES)

In collaboration with Transfaction Ltd., we have access to real scheduling data and manually-scheduled consignments for small UK hauliers (referred to as *real data*). The real data are insufficient, in quantity and quality, for our scheduling research, but provide us with indicative distributions and other information. By using the parameterised DES, we generate larger, realistic, data sets on requests and consignments (referred as *generated data*).

We use the real data and its distributions to parameterise a DES of the dynamic receipt of consignment requests. In particular, the arrival times of orders in the system is not recorded by manual schedulers, either for customer orders or for backhauls; the distributions for these are estimated from the typical pattern of diurnal and weekly orders.

4.2 The VNDM Heuristic

Like other hyper-heuristic approaches, VNDM is a two-phase approach, in which an initial set of routes is created and then optimised. The hyper-heuristic search framework uses a heuristic to select from a set of LLHs (Section 4.3, below).

In PMDP, each haulage company (carrier) is assumed to have an unlimited number of vehicles and is represented by a depot, randomly located within the area encompassing the consignments. A schedule consists of a number of routes, each beginning and terminating at one of the depots. To prepare an initial set of routes, consignments are sorted according to the time that they arrive in the system, and inserted greedily into a schedule, earliest arrival first. Consignments are assigned to a carrier from those carriers with fewest consignments that is geographically closest to the midpoint between a consignment's pickup and final delivery locations. Thus, the initial schedule systematically distributes consignments evenly across many carriers.

In order to simulate dynamic request arrivals, in which new consignments are added to a schedule that is already being serviced, we keep track of simulation time (an internal representation of current time, stored so that requests which in reality would have already happened cannot be modified by our optimisation procedure). If the scheduled start time of any request is before the current simulation time, it is marked as "fixed". Additional requests cannot be inserted before these fixed requests, and the routing of a fixed request cannot be altered in any optimising moves. The insertion heuristic treats consignments atomically, finding the lowest cost insertion location across all routes for a pickup and all its deliveries (guaranteeing LIFO), such that no previously inserted consignment incurs a delay. After the insertion of each new consignment, VNDM is used for optimisation, running for a constant amount of CPU time.

VNDM is a descent-based first-improvement heuristic. Routes are first ordered by length, then each LLH generates a list of potential moves. Since the majority of a schedule is unaltered after a modification, VNDM limits revisiting parts of the search space by maintaining a record of LLHs that give no improvement on each route (pairs of route and LLH identifiers are stored). If a LLH fails to produce an improving move, it is added to a tabu list. The tabu list is re-initialised when a route is subsequently modified, as a LLH may now be able to find improvement where none was previously possible.

VNDM differs from other published PDP solution approaches in a number of ways, notably in the choice of local moves used (specific to the PMDP), the use of route ordering to focus the search on promising areas, and the use of a route memory to reduce repeated searching. The search space is further reduced by imposing distance and time limits on nodes chosen for potential moves, which are different for each LLH and determined through extensive testing.

4.3 Low-level Heuristics

The nature of PMDP, with strict LIFO ordering of consignments, guides our selection of LLHs to apply to route optimisation. Since a pickup request must occur before its delivery requests, reversing a section of a schedule and repairing infeasible pickup / delivery ordering will significantly alter the distance of the route. Because time windows are usually tight, increased distance may result in significant delay in servicing requests.

Highly disruptive LLHs have been found incapable of improving our schedules, ruling out LLHs that use partial route inversions, such as GENI (Gendreau et al., 1992) and iCROSS (Bräysy, 2003). However, we can use the CROSS exchange of Savelsbergh (1992) (used by Taillard et al. (1997)) as it does not reverse chains of requests. Additional LLHs, such as GENI-PO (Mourdjis et al., 2014), have been chosen or developed to preserve existing schedule ordering as much as possible. By keeping the pickup and deliveries of one consignment in the same schedule (rather than splitting the consignment across loads and using precedence constraints), we facilitate the use of LLHs from the widely-researched area of one-many-one VRPs.

In selecting LLHs to modify routes, a consignment may only be rescheduled if the modification results in a valid schedule. A consignment may be scheduled such that other pickups or deliveries occur between the consignment's pickup and final delivery, providing load and LIFO constraints are not violated. However, if the consignment is rescheduled, the nested pickups or deliveries from other consignments remain in the original schedule, thus allowing modifications to undo nested consignments.

We provide four LLHs that can be applied to a single route. If a single route operator can generate more than one resulting route, that which is least disruptive to existing schedule ordering is used. LLHs that would reverse the order of a chain of requests are not allowed, hence we do not use 2-Opt.

3-Opt moves one consignment to a different position in the route schedule, whilst *4-Opt* swaps the positions of two consignments in the route schedule. *Nest Consignment* moves a whole consignment to a position within the delivery schedule of another consignment, thus nesting the first consignment within the second. Finally, *Nest Two Consignments* nests two consignments inside other consignments, a useful move where single-level nesting produces no improvement.

We provide four further LLHs that operate on more than one route at a time. GENI-PO (Mourdjis et al., 2014) is a non-inverting variant of GENI (Gendreau et al., 1992). The other three LLHs are from Savelsbergh (1992). *Relocate* moves one consignment to a valid position in a different route schedule, which may introduce nesting. *Geni-PO* is a variation of relocate that preserves as much previous ordering as possible by moving a consignment to be geographically close to other consignments: all possible insertion position pairs are considered to find the most improving relocation. *Swap* exchanges consignments from two different routes, whilst *Cross* exchanges two chains of consignments between routes, preserving the existing ordering within each chain. Cross considers chains of all lengths when used.

4.3.1 Use of Local Moves

Of the eight LLHs, three consume only small amounts of CPU time for problems of the size we study (3-Opt, 4-Opt and Nest consignment), whilst the others (Nest two consignments, Relocate, Geni-PO, Swap and Cross) are considered *hard* and take a significant amount of time. However, the hard LLHs generate several orders of magnitude more potential moves than the computationally trivial moves. There is no intuitive reason to prefer one hard LLH to another, and there is little advantage to running more than one hard LLH at a time. Thus, to prevent VNDM optimisation simply running out of time whilst applying too many hard LLHs, each call of VNDM uses a neighbourhood structure comprising the three low-CPU LLHs in the order above, then one hard LLH, selected at random. The random selection ensures that all the hard LLHs are used over a series of optimisations, and thus provides ample diversification.

5 COMPUTATIONAL RESULTS

In order to investigate the effects of cooperation, we generate 100 scenarios from data on a set of 27,153

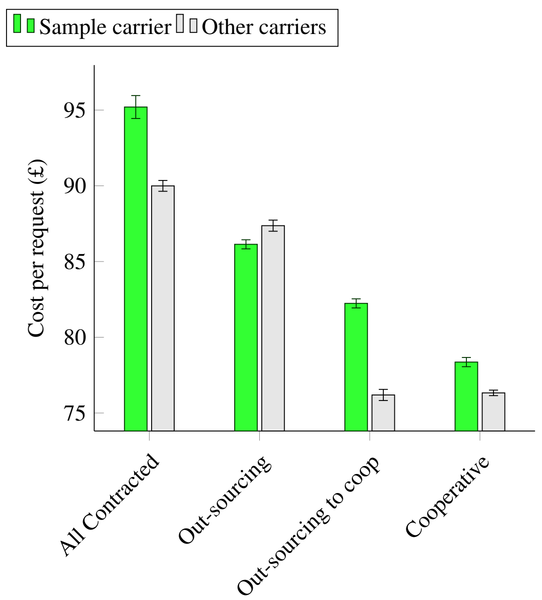

Figure 1: Average savings for a single carrier (sample carrier) and a group of carriers (other carriers), with four different models of cooperation.

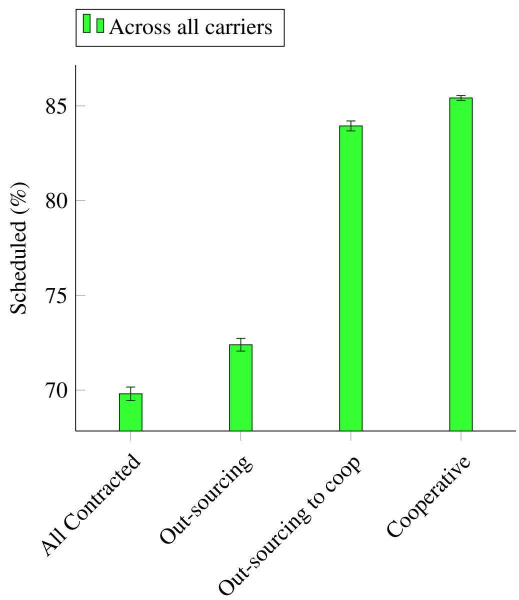

Figure 2: Percentage of contracted consignments serviced across the four different models of cooperation.

real-world consignments. The scenarios are built by selecting 200 real consignments at random from this set and building pairs of consignments representing outbound *linehaul* and return *backhaul* legs. Each consignment consists of at least two requests.

The scenarios examined use five carriers, and explore the effect on one carrier (the sample) under four different configurations of cooperation with the other four carriers. *All Contracted* has all consignments assigned to a single carrier. Optimisation is only possible between vehicles belonging to the same carrier. *Out-sourcing* starts with a contracted model, but allows re-assignment of consignments from the sample to any of the other carriers, if cost savings can be made. *Out-sourcing to coop(erative)* adds the out-sourcing model for the sample carrier into a model in which the other carriers can exchange consignments if savings can be made; the sample carrier does not accept any additional consignments. Finally, the *cooperative* model initially assigns all consignments to individual carriers (as in *Contracted*) but allows unrestricted re-allocation during optimisation, if cost savings are possible.

We simulate one dynamic scheduling week, and limit optimisations to 5 minutes of CPU time. Each of the four configurations is run 30 times, using a heterogeneous cluster of Intel Xeon based servers, totalling 72 cores and 120GB of RAM. The results presented here thus represent one thousand hours of CPU time.

Figure 1 shows that for the sample carrier, an average 9% saving can be made by out-sourcing to the four other carriers, whilst the configuration that allows other carriers to also cooperate results in average savings of nearly 14%, because the cooperation allows more efficient routing across the carriers. If the sample carrier also cooperates in efficient scheduling, the total average saving for the sample carrier rises to 18%. Cooperation is also beneficial for the other carriers: accepting orders from the single carrier can produce benefits of 3%, whilst cross-group cooperation produces savings to averaging 15%.

The results shown should drive all carriers towards cooperation. Competition favours carriers with the lowest costs; the sample carrier achieves this in configuration 2, by outsourcing to other carriers who are not cooperating. However, rational competitors would be expected to copy this behaviour, moving the system towards a reallocation of consignments as seen in configuration 3; here, the competitors are cooperating, and the sample carrier is at a competitive disadvantage. However, if all carriers cooperate, as in configuration 4, the lowest costs for all carriers are observed.

Increasing cooperation allows a greater number of consignments to be handled. Figure 2 shows that the schedule in which all carriers operate alone covers on average less than 70% of their contracted consignments. However, the fully cooperative model can schedule over 85% of consignments. (Note that random scenario generation means that there is no guarantee that all consignments are feasible given the number of carriers, their locations and that even with an infinite number of vehicles, some consignments are too far apart to be serviced whilst adhering to driver working hour rules: since we do not consider driver sleeping arrangements and all routes must begin and end at the depot, these consignments are impossible in our current model.)

Table 1: Percentage of the sample carrier's consignments re-allocated in different configurations.

Config.	Cooperation Mode	Re-allocated
1	All contracted	0%
2	Out-sourcing	65.6%
3	Out-sourcing to coop	67.2%
4	Cooperative	57.2%

Table 1 shows the percentage of consignments that are re-allocated from the sample carrier in each configuration. Both out-sourcing and out-sourcing to a cooperative allow almost two-thirds of the carrier's consignments to be assigned to others: because our scheduling algorithm minimises cost, these re-allocations can be interpreted as being carried more cheaply, due to more efficient use of resources, when assigned to other carriers. We are most interested in the percentage of consignments that are re-allocated away from the sample carrier. When outsourcing and cooperation are combined (configuration 3), the sample carrier's re-assigned loads are most cost-effective, as, in this configuration, the other carriers can also re-allocate loads among themselves (but not to the sample carrier). In the fully cooperative model, the sample carrier's consignments are less cost-effectively reassigned than in other reallocation configurations. However, the overall cost-effectiveness of the 5 carriers is significantly better than in other configurations: 62.5% of other carriers' consignments were reallocated in this model, leading to the reduction in cost observed for cooperation in figure 1. These results also strongly support the contention that savings can accrue to small hauliers who cooperate to carry each others' consignments efficiently.

5.1 More Group Configurations

We seek to further investigate the effects of different sized groups of carriers on both cost and network capacity.

Using the same 100 scenarios as investigated previously, we now investigate how efficiently 10 carriers can service the consignments, split into a number of different group configurations. Cooperation is allowed within but not between these groups. In the *All Contracted* configuration each of the 10 carriers works independently, in the second configuration, carriers work in *Pairs*. in *1 vs 3s*, one carrier, the sample, is compared against 3 groups of 3 carriers. In *5 vs 5*, *3 vs 7* and *1 vs 9* the 10 carriers are divided into 2 groups of differing sizes accordingly. In the final configuration, *All Cooperative*, the 10 carriers work together.

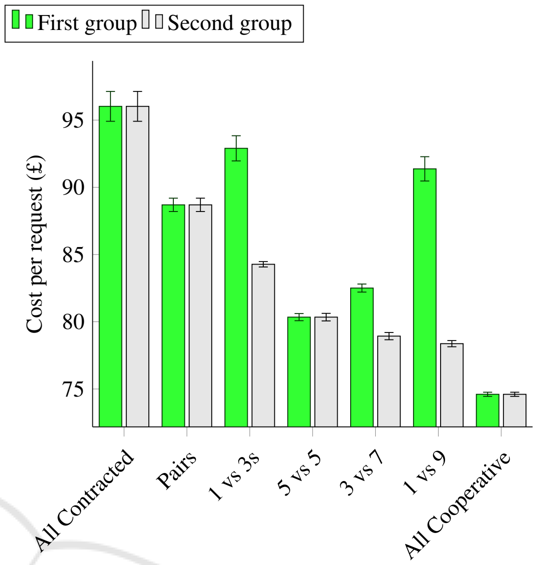

Figure 3: Cost per request for different carrier group configurations.

Figure 3 confirms our earlier findings that working as a group can substantially reduce costs and additionally shows that larger groups can attain bigger cost reductions than smaller groups.

In each configuration, consignments are divided equally between groups, not carriers, such that, for example in the *1 vs 3s* configuration each group of carriers is assigned 100 consignments out of 400 but in the *1 vs 9* configuration, each group is assigned 200 consignments. Because of this, carrier 1 has more choice in the 1 vs 9 configuration and can achieve slightly better results than in the 1 vs 3s configuration, however the number of consignments that can actually be served is dramatically reduced as can be seen in figure 4.

Figure 4 shows again the increase in network capacity made possible through cooperation. It is also clear that the largest savings are made quickly: just pairing with one other carrier can increase the number of scheduled deliveries from 72% to 80%.

5.2 Carrier Group Size

Extending our analysis, we seek to identify if there are diminishing returns for increasing the number of carriers in a cooperative group.

Figure 5 shows how both the cost per request and the percentage of consignments scheduled improve

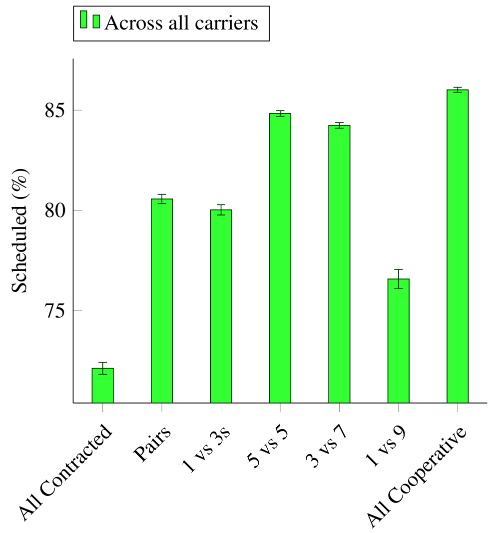

Figure 4: Percentage of scheduled consignments for different carrier group configurations.

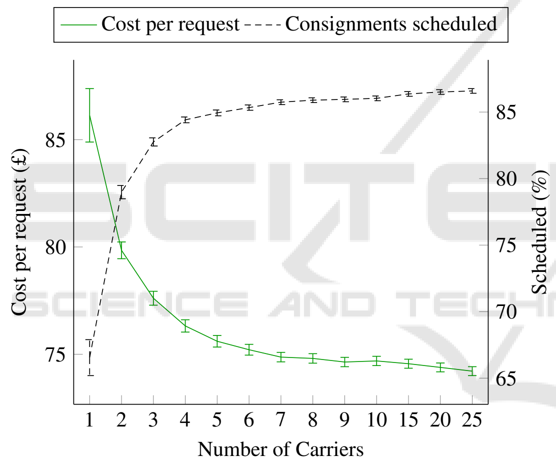

Figure 5: Cost per request and number of consignments successfully scheduled as the number of carriers working together increases.

as the number of carriers in a cooperative group increases.

Though there are linear savings evident above 10 carriers the majority of benefit is found between 1 and 5 carriers. These results must be qualified by stating that our consignments cover the UK and our carriers are randomly located across this area; since distance costs are a dominant factor in real-world pricing; if larger distances are involved, for instance across Europe, America or Asia a larger number of well distributed carriers would likely be necessary to produce these savings. We also assume an infinite number of vehicles at each carrier location; in practice there will be a limited supply of vehicles at each carrier and therefore multiple carriers in the same area would need to work together. Our results can be thought of more as suggesting that 10 major transport hubs is sufficient for efficient vehicle routes in the UK assuming an unlimited number of vehicles, clearly, given that there are over 6000 carriers in the UK, more than 10 carriers would be required.

6 DISCUSSION AND CONCLUSIONS

We have presented the VNDM hyper-heuristic, and a set of LLHs optimised for the ordering constraints of the problem, as an effective schedule optimisation for PMDP under dynamic consignment requests. We use data from the RHA to explore pricing and marginal costs of consignments, and show that cost savings of 15% to 18% are possible when hauliers cooperate. Cooperation also increases the capacity of a group of hauliers, by as much as 21%. The benefits of cooperation see diminishing returns above 10 separate carrier locations working together assuming sufficient numbers of vehicles to meet demands. Larger cooperatives will always have lower operating costs than smaller ones as they are able to more efficiently schedule their consignments to existing vehicle routes.

In practice there would need to be some way to distribute savings amongst all cooperating carriers fairly, in order to encourage participation. Further work is required to determine the best way to do this, potential strategies involve allowing carriers to auction jobs to cooperating parties or having some central control involved in deciding which carriers take what consignments. We also do not consider issues of vehicle reliability, for example, who pays the costs associated with missed delivery slots and what effect this has on customer perceptions. We intend to investigate the impact of planning time and arrival time windows on consignment cost and also to perform a comparison with alternative hyper-heuristic approaches.

We have not considered the fixed costs associated with carrier owned vehicles in this research, only savings to marginal costs of delivery. Implementing the strategies outlined in this paper may result in reduced usage of carrier owned assets; we see future work as being able to advise on appropriate fleet sizes for carriers and note that in general, cooperation allows for an increase in capacity, allowing the same fixed cost assets to be more productive, assuming there is sufficient demand for service.

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

Albareda-Sambola, M., Fernández, E., and Laporte, G. (2014). The dynamic multiperiod vehicle routing problem with probabilistic information. *Computers & Operations Research*, 48(1):31–39.

Belhaiza, S., Hansen, P., and Laporte, G. (2013). A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. *Computers & Operations Research*, 52 part B:269–281.

Benavent, E., Landete, M., Mota, E., and Tirado, G. (2015). The multiple vehicle pickup and delivery problem with LIFO constraints. *European Journal of Operational Research*, 243(3):752–762.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1):1–31.

Berbeglia, G., Cordeau, J.-F., and Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1):8–15.

Bräysy, O. (2003). A Reactive Variable Neighborhood Search for the Vehicle-Routing Problem with Time Windows. *INFORMS Journal On Computing*, 15(4):347–368.

Bräysy, O. and Gendreau, M. (2005a). Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science*, 39(1):104–118.

Bräysy, O. and Gendreau, M. (2005b). Vehicle Routing Problem with Time Windows, Part II: Metaheuristics. *Transportation Science*, 39(1):119–139.

Cherkesly, M., Desaulniers, G., and Laporte, G. (2015). Branch-Price-and-Cut Algorithns for the Pickup and Delivery Problem with Time Windows and LIFO Loading. *Computers & Operations Research*, 62(1):23–35.

Clarke, G. and Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4):568–581.

Crainic, T. G., Nguyen, P. K., and Toulouse, M. (2015). Synchronized Multi-Trip Multi-Traffic Pickup & Delivery in City Logistics. *CIRRELT*, 05(February):1–24.

Demir, E., Bekta, T., and Laporte, G. (2014). A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793.

Desaulniers, G., Desrosiers, J., Solomon, M. M., Erdmann, A., and Soumis, F. (2002). VRP with pickup and delivery. In Toth, P. and Vigo, D., editors, *The vehicle routing problem*, pages 225–242. SIAM.

Dff International Ltd, R. (2014). *RHA Cost Tables*. Retrieved from http://www.rha.uk.net/.

Dff International Ltd, R. (2015). *RHA National Directory of Hauliers*. Retrieved from http://www.rha.uk.net/.

Dumas, Y., Desrosiers, J., and Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54(1):7–22.

Gendreau, M., Guertin, F., Potvin, J.-Y., and Séguin, R. (2006). Neighborhood Search Heuristics for a Dynamic Vehicle Dispatching Problem with Pick-ups and Deliveries. *Transportation Research Part C: Emerging Technologies*, 14(3):157–174.

Gendreau, M., Hertz, A., and Laporte, G. (1992). New Insertion and Post Optimization Procedures for the Traveling Salesman Problem. *Operations Research*, 40(6):1086–1095.

Gschwind, T., Irnich, S., and Mainz, D. (2012). Effective Handling of Dynamic Time Windows and Synchronization with Precedences for Exact Vehicle Routing. Technical report, Johannes Gutenberg University Mainz, Mainz, Germany. Retrieved from http://logistik.bwl.uni-mainz.de/.

Laporte, G. (2009). Fifty Years of Vehicle Routing. *Transportation Science*, 43(4):408–416.

McLeod, F., Cherrett, T., Shingleton, D., Bekta, T., Speed, C., Davies, N., Dickinson, J., and Norgate, S. (2012). 'Sixth Sense Logistics: Challenges in supporting more flexible, human-centric scheduling in the service sector. In *Annual Logistics Research Network (LRN) Conference*, Cranfield, UK.

Mitrović-Minić, S., Krishnamurti, R., and Laporte, G. (2004). Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, 38(8):669–685.

Mourdjis, P. J., Cowling, P. I., and Robinson, M. (2014). Metaheuristics for the pick-up and delivery problem with contracted orders. In Blum, C. and Ochoa, G., editors, *Evolutionary Computation in Combinatorial Optimization*, pages 170–181, Berlin Heidelberg. Springer-Verlag.

Nahum, O. E. (2013). *The Real-Time Multi-Objective Vehicle Routing Problem*. Phd, Bar-Ilan University.

Rochat, Y. and Taillard, É. D. (1995). Probabilistic Diversification And Intensification In Local Search For Vehicle Routing. *Journal of Heuristics*, 1(1):147–167.

Savelsbergh, M. W. P. (1992). The Vehicle Routing Problem with Time Windows: Minimizing Route Duration. *INFORMS Journal on Computing*, 4(2):146–154.

Taillard, É. D., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J.-Y. (1997). A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. *Transportation Science*, 31(2):170–186.

Toth, P. and Vigo, D. (1997). Heuristic Algorithms for the Handicapped Persons Transportation Problem. *Transportation Science*, 31(1):60–71.

Xu, H., Chen, Z.-L., Rajagopal, S., and Arunapuram, S. (2003). Solving a Practical Pickup and Delivery Problem. *Transportation Science*, 37(3):347–364.

APPENDIX

Model Constraints

The constraints for the PMDP are laid out in tables 2 and 3. The constraints in table 2 have been adapted and expanded from the formulation for the PDP presented by Desaulniers et al. (2002); Table 3 presents the additional new constraints for the PMDP.

Table 2: Adapted constraints from Desaulniers et al. (2002), here $\equiv$ implies that this constraint is equivalent to a constraint presented by Desaulniers et al. and $*$ implies that this constraint has been modified for the PMDP.

$\equiv$	$\sum_{k\in K}\sum_{u\in R_k} b_{ruk} = 1$ $\forall r \in R$	(3)
$*$	$\sum_{u\in P_k} b_{ruk} * \lvert D_j\rvert - \sum_{w\in D_j} b_{rwk} = 0$ $\forall k \in K, r \in R_k$	(4)
$*$	Removed	(5)
$*$	Removed	(6)
$*$	Removed	(7)
$\equiv$	$b_{ruk}\left(t_{rk} + tt_r^{service} + tt_{ru} - t_{uk}\right) \leq 0$ $\forall k \in K, (r,u) \in A_k$	(8)
$*$	$t_r^{start} \leq t_r^{end}, t_r^{start} \leq t_{rk}$ $\forall k \in K, r \in R_k$	(9)
$*$	$t_{rk} + tt_r^{service} + tt_{ru} \leq r_{uk}$ $\forall k \in K, r \in P_k, u \in D_r$	(10)
$\equiv$	$b_{ruk}\left(ll_{rk} + l_u - ll_{uk}\right) = 0$ $\forall k \in K, (r,u) \in A_k$	(11)
$*$	$0 < l_r \leq ll_{rk} \leq l_k$ $\forall k \in K, r \in P_k$	(12)
$*$	$l_r + \sum_{u\in D_r} l_u = 0$ $\forall r \in P$	(13)
$\equiv$	$l_o(k) = 0$ $\forall k \in K$	(14)
$\equiv$	$b_{ruk} \geq 0$ $\forall k \in K, (r,u) \in A_k$	(15)
$\equiv$	b_{ruk} binary $\forall k \in K, (r,u) \in A_k$	(16)

Constraints (3) and (4) ensure that each arc is only included once and that a pickup and all its corresponding deliveries are handled by the same truck. Here, $|D_u|$ is the number of delivery-requests for pickup-request u. Constraint (4) is non-standard for the PDP and is necessary as there may be multiple delivery-requests per pickup-request. It states that for each pickup request there exists an $b_{ruk} = 1$ and that this multiplied by the number of deliveries is the same as the number of arcs that end at each of the corresponding delivery requests. Constraints (5) to (7) have been removed in comparison to the formulation in Desaulniers et al. (2002) as we do not have a multicommodity flow. Constraint (8) remains unchanged however, (9) and (10) have been modified, together these allow for soft time windows. Constraints (11) to (13) specify that a pickup node must have positive load and that deliveries must have negative load, also that the sum of pickup and delivery loads is zero. The initial vehicle load, non-negativity and binary requirements are the same as Desaulniers et al. (2002). The following constraints have been added for the PMDP: (17) and (18) specify that a request has exactly one pickup and may have arbitrarily many deliveries. (19) specifies the precedence between a pickup and its deliveries while (20) expresses the LIFO constraint. Finally, (21) specifies that the length (in time) of any vehicles route is less than a value E_k which may be set according to local conditions.

Table 3: New constraints for the PMDP.

$\lvert P_c\rvert = 1$ $\forall i \in I$	(17)
$\lvert D_c\rvert \geq 1$ $\forall i \in I$	(18)
$t_{rk} < t_{uk}$ $\forall k \in K, r \in P_k, u \in D_r$	(19)
$t_{rk} < t_{uk} \Rightarrow t_{vk} < t_{wk}$ $\forall k \in K,\ \forall r,u \in P_k,\ \forall v \in D_u,\ \forall w \in D_r$	(20)
$\sum_{(r,u)\in A_k} b_{ruk}\left(tt_r^{service} + tt_{ru}\right) \leq tt_k$ $\forall k \in K$	(21)

Minimising k, the number of vehicles used, is not considered as part of this problem, though it is kept low as a side effect of the heuristics used. For each truck, requests may be nested within other requests if LIFO and capacity constraints are not violated.