

Towards Collaborative Optimisation in a Shared-Logistics Environment for Pickup and Delivery Operations

Timothy Curtois¹, Wasakorn Laesanklang¹, Dario Landa-Silva¹, Mohammad Mesgarpour² and Yi Qu¹

¹*ASAP Research Group, School of Computer Science, The University of Nottingham, Nottingham NG8 1BB, UK*

²*Microlise Ltd, Eastwood, Nottingham, NG16 3AG, UK*

*{tim.curtois, wasakorn.laesanklang, dario.landasilva, yi.qu}@nottingham.ac.uk,
mohammad.mesgarpour@microlise.com*

Keywords: VRP, PDP, LNS, Horizontal collaboration, Split-loads.

Abstract: This paper gives an overview of research work in progress within the COSLE (Collaborative Optimisation in a Shared Logistics Environment) project between the University of Nottingham and Microlise Ltd. This is an R&D project that seeks to develop optimisation technology to enable more efficient collaboration in transportation, particularly real-world operational environments involving pickup and delivery problems. The overall aim of the project is to integrate various optimisation techniques into a framework that facilitates collaboration in a shared freight transport logistics environment with the overall goal of reducing empty mileage.

1 INTRODUCTION

This paper provides an overview of the research work being undertaken as part of the COSLE (Collaborative Optimisation in a Shared Logistics Environment) project. This is an R&D project between the University of Nottingham and Microlise Ltd in the UK. The overall objective of the project is to develop optimisation technology to enable more efficient collaboration in transportation. According to a recent report from the Institution of Mechanical Engineers in 2016 (Oldham, 2016), up to 30% of all commercial vehicles on UK roads travel empty, which leads to around 150m wasted road miles, 200,000 additional truck journeys, increased road congestion and about 200,000 tonnes of unnecessary CO₂ emissions. One way to improve this situation is by facilitating collaboration between carriers. The improved cooperation will reduce the total distances that vehicles travel without loads (so called empty miles), increase vehicle utilisation metrics and decrease distribution costs. Already there have been several successful applications of increased cooperation in transportation. Crujssen et al. (2007a) modelled a joint route planning problem, used a benchmark case and reported that 30.7% of total distribution costs are saved. Ergun et al. (2007) proposed a lane covering problem and solved it using heuristics. Results showed that the saving range from about 5.5% to a little over 13%, again using real data. Frisk et al. (2010) presented a

case study of horizontal collaboration in tactical transportation planning between eight forest companies and the results showed up to 14.2% of the transportation cost saved. Pérez-Bernabeu et al. (2015) discussed horizontal collaboration in road transportation and presented numerical analysis based on a set of well-known benchmarks for the Multidepot Vehicle Routing Problem. The average cost reduction ranges from 5% to 90% depending on the geographical distribution of customers with respect to their transport service providers. It has been proved by researchers that saving of costs and increase of resource utilization can be attained throughout horizontal collaboration (Crujssen et al., 2007b; Wang and Kopfer, 2014).

The goal of the COSLE project is to develop an innovative service to enable collaboration in a shared freight transport logistics environment to reduce empty freight runs. As part of this, three sub-projects related to scheduling and optimisation have been identified and are currently being undertaken by project team. This position paper provides an overview of these sub-projects and the progress achieved so far. The first sub-project is to develop a methodology to tackle pickup and delivery problems with time windows and other real-world constraints. A metaheuristic approach has been developed and tested on a benchmark data set. This work and a summary of the results is described further in Section 2. The second sub-project is a method to enable optimal load-splitting within routes and sched-

ules. This is presented in Section 3 as well as a review of related research. The third sub-project, discussed in Section 4, is to develop a methodology for assigning new customers within existing routes. The outcomes from these three sub-projects will be integrated into a framework that will aim to enable more efficient collaboration in transportation under real-world operational conditions. The overall approach is to develop an optimisation engine for tackling pickup and delivery routing scenarios in which various transportation operators are willing to collaborate in order to increase the overall utilisation of vehicles by reducing the number of empty runs.

2 A HYBRID METAHEURISTIC FOR PDP

The pickup and delivery problem (PDP) is a widely occurring vehicle routing problem. Similar to other vehicle routing problems it often contains window and capacity constraints. Unlike the general vehicle routing problem however PDP also includes pairing and precedence constraints. The pairing constraint is to ensure that a pickup customer and its associated delivery customer are both serviced by the same vehicle. The precedence constraint does not allow a delivery customer to be visited before its associated pickup customer. The techniques being investigated in this project are for tackling pickup and delivery problems which also contain window and capacity constraints as well as other real world constraints such as driver working time regulations and break requirements. The objective function, similar to other problems, requires the minimisation of the number of vehicles used and the total distance travelled. Other optional objectives allow the minimisation of total driver hours, and a profit maximisation objective for problems in which some customers may be optionally serviced and have an associated completion cost.

Various heuristic and exact methods have been proposed for PDP. Each method has advantages and disadvantages. The exact methods, although extremely effective on smaller instances, appear to still be difficult to apply to the largest instances. Metaheuristics however have been shown to scale much more easily to larger instances although are easily beaten on smaller instances. They can also provide no information on solution optimality or even bounds. However a recent survey (Hall and Partyka, 2016) suggests that most industrial vehicle routing packages are still heavily biased towards using metaheuristics.

Of the exact methods published, many are versions of the column generation and branch and price

framework (Dumas et al., 1991; Ropke and Cordeau, 2009; Savelsbergh and Sol, 1998; Venkateshan and Mathur, 2011; Xu et al., 2003) or less commonly, branch and cut (Lu and Dessouky, 2004; Ruland and Rodin, 1997). Examples of metaheuristics include Bent and Van Hentenryck (2006); Li and Lim (2003); Nagata and Kobayashi (2010); Nanry and Barnes (2000); Ropke and Pisinger (2006). Metaheuristics have also been applied to less common variants of PDP (Cherkesly et al., 2015; Kammarti et al., 2004; Masson et al., 2013). Several survey papers are also available (Berbeglia et al., 2007; Parragh et al., 2008; Savelsbergh and Sol, 1995).

The hybrid method that has been developed here combines Local Search, Large Neighbourhood Search (LNS) and Guided Ejection Search (GES). It works in several phases. In the first phase, local search using four different neighbourhood operators is used to create an initial solution. The operators used are:

1. Inserting unassigned customers into routes.
2. Moving a customer from one route to another.
3. Swapping customers between routes.
4. Moving a customer from one route to a second route and simultaneously moving a customer from a second route to a third route.

Even on the largest instances the local search phase is very fast but the solutions produced are nearly always quite sub-optimal. The next phase uses Guided Ejection Search in an attempt to minimise the total number of routes in the solution. The GES implementation is based on Nagata and Kobayashi (2010). It works by iteratively, randomly selecting a route, un-assigning all customers in the route and then attempting to re-insert these customers in existing routes. If it is unable to insert a customer it ejects one or more customers from a route to enable it to insert the customer. The ejected customer(s) are then added to the list of customers still to be inserted. After the ejection the solution is perturbed by randomly moving or swapping already assigned customers between routes. This helps to insert un-assigned customers and also prevents infinite loops. This is repeated for a number of iterations or until all customers have been inserted. If all customers are inserted then the process is repeated to try and remove another route. Otherwise the original solution is restored.

After the GES phase, a Large Neighbourhood Search is applied to try and improve the other objectives (total distance for the benchmark instances). The LNS is a simplified version of the adaptive LNS of Ropke and Pisinger (2006). One of the simplifi-

Table 1: Summary of Results

Customers	Instances	New best knowns	Equal best knowns
50	56	0	56
100	60	7	35
200	60	22	19
300	60	33	6
400	60	45	5
500	58	35	4

cations was to remove the adaption procedure which was shown by the authors to have only a small percentage benefit. Our results also confirmed that excellent solutions could still be obtained without the adaption procedure. Another modification was to replace a simulated annealing heuristic with a late acceptance hill climbing heuristic (Burke and Bykov, 2012). The motivation for this was to remove the number of parameters that required setting. The algorithm operates by iteratively un-assigning a small number of customers and then heuristically attempting to re-insert them but in a lower cost configuration. If it is unable to re-insert them in a better way then it restores the original solution and selects a new set of customers for removal and re-insertion. This process is iteratively repeated. The customers for removal are selected randomly or via the Shaw heuristic (Shaw, 1998) which selects customers that are similar in terms of location, time windows and order size. The insertion heuristics are based on the regret assignment heuristic (Ropke and Pisinger, 2006).

After the completion of the LNS phase, if there is time remaining then the best known solution is perturbed by randomly moving or swapping customers between routes. The three phases are then applied again to the perturbed solution. This whole process is repeated until a fixed time limit is reached. At which point the best known solution is returned.

In order to evaluate the efficacy of the algorithm it was applied to the well-known benchmark problem instances of Li and Lim (2003)¹. These instances are divided by size into six groups, ranging from 50 customers up to 500 customers. The results are summarized in Table 1. In Table 1, the column “New best solution” indicates the number of instances that our algorithm was able to find a new best known solution. The column “equal best knowns” indicates the number of instances on which the algorithm was able to equal the current best known solution for that instance.

¹Available at <http://www.sintef.no/projectweb/top/pdptw/li-lim-benchmark/>

3 LOAD SPLITTING

Often job loads can be partially collected and delivered multiple times provided they are completed in entirety within the given time windows. This option allows the possibility of split loads to be used operationally.

The obvious case is when a requested demand exceeds vehicle capacity. Demands in this case must be split before the optimisation process. The research question in this case is how to split the requested demands so that the split loads aid the optimisation process. Some of the papers in the literature saw this case as a part of pre-processing in optimisation problem.

The other case is to gain additional savings in the operational plan. The literature shows split loads can reduce operational costs (Andersson et al., 2011; Nowak et al., 2009). The benefits of split loads were subject to problem characteristics such as load size, stopping cost, and frequency of loads having common pickup and delivery locations.

The transportation problem with split loads arose in the generic vehicle routing problem (VRP) in (Dror and Trudeau, 1989). The idea was to relax the VRP so that a customer can be visited more than once. A k -split interchange is also proposed as a heuristic procedure to split a demand into multiple loads. The solution after the split procedure was expected to have cost reduction from the generic problem.

Dror and Trudeau (1989) used a heuristic process to tackle a split load problem in two stages: construct a solution to the generic VRP; and apply k -split interchange and improvement routine to get a solution to the split load problem. The k -split interchange was also used as a move in a tabu search algorithm for the split delivery vehicle routing problem by Archetti et al. (2006). They describe k -split interchange in two main procedures:

1. Remove a demand i from all routes where it is visited; and
2. Find a route subset R where the summation of remaining capacity is larger than the demand i so that the demand i is split into every route in the route subset R .

The route subset R should have the least insertion cost.

A randomised granular tabu search heuristic was used to solve the split delivery vehicle routing problem by Berbotto et al. (2014). The method builds a granular neighbourhood to reduce the computational time required to explore solution neighbourhood.

Pickup and delivery with split loads was tackled by a heuristic where two route segments of different routes can visit the same pickup and delivery demand

by Nowak et al. (2008). The demand can be carried by two vehicles.

A requirement of split delivery in simultaneous pickup and delivery arose in automobile industries in Tang et al. (2009). At the supplier location, a truck must deliver empty bins to the pickup locations in order to pick up the full bins. In the same way, at the manufacturer, the truck must deliver full bins and pick up empty bins. Bins are cycled between the manufacturer and the suppliers.

Coordination split delivery can also benefit retailers in maintaining stock levels (Li et al., 2011). This approach can reduce retailer inventory costs while the transportation cost remains the same. A similar application can apply to the natural disaster relief distribution problem (Wang et al., 2014). The goal of this case was to distribute sufficient aid to the disaster areas. A disaster area can be visited multiple times. A full review on split delivery transportation problems can also be found in Archetti and Speranza (2012).

The closest application to the split pickup and delivery in the collaborative logistics environment considered here is the pickup and delivery problem with split load proposed by Nowak et al. (2008). Therefore, we adopt their split load creation procedure and apply it to the large neighbourhood search method. The same procedure can also be used to split demand that exceeds vehicle capacity. The split load creation procedure works similarly to the k -split interchange procedure. The procedure is as follows:

1. Find a segment i to split;
2. Find a segment j the load should move to;
3. Split the load in the segment i where the first split load is equal to the excess capacity of segment j , and the second split load is the remainder;
4. Move the split load to segment j and the remainder load is kept in the segment i ;
5. Perform the search heuristic.

The proposed approach in this sub-project is to implement move operators within LNS in order to handle split loads. This includes the delete split operator, exchange split operator, etc. These operators were applied to VRP (Berbotto et al., 2014). The delete split operator removes a set of split loads to become a full demand load.

The exchange split operator swaps the load splitting position in a selected route. In VRP, the idea was to swap the position to split a load while maintaining vehicle capacity. Suppose we have a load i and a load j where load i is split into two smaller loads and load j and one of the split loads of i are assigned to a vehicle. This operator relocates the split position

from load i to load j which results in the vehicle taking the full load i and a partial load j . For our PDP, the operator starts from selecting an interval where a vehicle has split loads in their fill. The exchange split operator will:

1. Delete the split of a demand; and
2. Apply a split to one of the other demands.

The demands that the exchange split operator can select must be the fill in the selected interval only. The operator keeps the visit order of the selected route but may change the order of the routes that operate on the split demands.

This splitting heuristic and move operators will be integrated into the hybrid metaheuristic for PDP outlined in Section 2 and applied to large real-world instances. The benefits of providing splitting options will then be analysed. Optionally, the splitting heuristic can be adapted to increase vehicle empty space available for taking advantage of new collaborative opportunities. In the same way, the heuristic can split the collaborative jobs so that they can be inserted into the existing routes.

4 CUSTOMER INSERTION INTO EXISTING ROUTES

Another requirement in collaborative transport operations is to be able to insert new customers into existing routing plans. It might be that the ordering of the customers in the routes of the existing solution cannot be changed but their arrival times could be adjusted provided that their window constraints are still respected. Existing customers must also remain within their current routes. Hence, the hybrid LNS+GES algorithm outlined in Section 2 cannot simply be applied to a new instance which includes the new customers. Instead, a separate mechanism is being developed to insert the new customers.

To the best of our knowledge this problem has little or no previously published research articles. Modified but similar versions of the problem do sometimes appear as sub-problems in methodologies for vehicle routing problems though. For example, related problems are solved using branch and bound and constraint programming algorithms in Bent and Van Hentenryck (2006); Shaw (1998). In Ropke and Pisinger (2006) also use a heuristic method to solve a version of the sub-problem for pickup and delivery with time window problems.

For the insertion problem considered here two separate objectives for two different scenarios are proposed:

1. Maximise the number of customers inserted.
2. Maximise profit. In this scenario customers are assigned values (revenue) and a cost is calculated based on total solution distance and/or total driver hours.

The insertion problem can be formulated as an integer programming problem and solved using a mathematical programming solver. We will also be investigating and comparing heuristic methods and alternative exact methods to establish the computation time/efficiency trade-off for the different approaches. For example, the hybrid LNS+GES algorithm already contains an existing insertion algorithm in the form of the regret heuristic. Greedy insertion heuristics are also feasible options. Other insertion algorithms that we will be developing and testing are the branch and bound approaches in Shaw (1998) and Bent and Van Hentenryck (2006).

To analyse the algorithms a testing framework has been created to allow us to efficiently repeat and reproduce the results. The test instances were created by taking existing instances, removing sets of customers and solving the reduced instances using the LNS+GES algorithm. The original instance is then used with this initial solution to form a new insertion instance. This procedure is repeated with different parameter settings to generate a large set of test instances to apply the algorithms to.

Another motivation for developing and analysing several insertion methods is to investigate whether a more efficient and effective method can be developed for the LNS algorithm. If so then it is possible that the LNS algorithm can be further improved by incorporating the new insertion algorithm.

ACKNOWLEDGEMENTS

We thank Innovate UK for funding the COSLE project (grant 102037).

REFERENCES

- Andersson, H., Christiansen, M., and Fagerholt, K. (2011). The maritime pickup and delivery problem with time windows and split loads. *INFOR: Information Systems and Operational Research*, 49(2):79–91.
- Archetti, C. and Speranza, M. G. (2012). Vehicle routing problems with split deliveries. *International transactions in operational research*, 19(1-2):3–22.
- Archetti, C., Speranza, M. G., and Hertz, A. (2006). A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, 40(1):64–73.
- Bent, R. and Van Hentenryck, P. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33(4):875–893.
- 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.
- Berbotto, L., García, S., and Nogales, F. J. (2014). A randomized granular tabu search heuristic for the split delivery vehicle routing problem. *Annals of Operations Research*, 222(1):153–173.
- Burke, E. K. and Bykov, Y. (2012). The late acceptance hill-climbing heuristic. *University of Stirling, Tech. Rep.*
- Cherkesly, M., Desaulniers, G., and Laporte, G. (2015). A population-based metaheuristic for the pickup and delivery problem with time windows and lifo loading. *Computers & Operations Research*, 62:23–35.
- Crujssen, F., Bräysy, O., Dullaert, W., Fleuren, H., and Salomon, M. (2007a). Joint route planning under varying market conditions. *International Journal of Physical Distribution & Logistics Management*, 37(4):287–304.
- Crujssen, F., Dullaert, W., and Fleuren, H. (2007b). Horizontal cooperation in transport and logistics: a literature review. *Transportation journal*, pages 22–39.
- Dror, M. and Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science*, 23(2):141–145.
- 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.
- Ergun, O., Kuyzu, G., and Savelsbergh, M. (2007). Reducing truckload transportation costs through collaboration. *Transportation Science*, 41(2):206–221.
- Frisk, M., Göthe-Lundgren, M., Jörnsten, K., and Rönnqvist, M. (2010). Cost allocation in collaborative forest transportation. *European Journal of Operational Research*, 205(2):448–458.
- Hall, R. and Partyka, J. (2016). Vehicle routing software survey: Higher expectations drive transformation. *ORMS-Today*, 43(1).
- Kammarti, R., Hammadi, S., Borne, P., and Ksouri, M. (2004). A new hybrid evolutionary approach for the pickup and delivery problem with time windows. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 2, pages 1498–1503. IEEE.
- Li, H. and Lim, A. (2003). A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools*, 12(02):173–186.
- Li, J., Chu, F., and Chen, H. (2011). Coordination of split deliveries in one-warehouse multi-retailer distribution systems. *Computers & Industrial Engineering*, 60(2):291–301.

- Lu, Q. and Dessouky, M. (2004). An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science*, 38(4):503–514.
- Masson, R., Lehuédé, F., and Péton, O. (2013). An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355.
- Nagata, Y. and Kobayashi, S. (2010). Guided ejection search for the pickup and delivery problem with time windows. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 202–213. Springer.
- Nanry, W. P. and Barnes, J. W. (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, 34(2):107–121.
- Nowak, M., Ergun, O., and White, C. C. (2009). An empirical study on the benefit of split loads with the pickup and delivery problem. *European Journal of Operational Research*, 198(3):734–740.
- Nowak, M., Ergun, Ö., and White III, C. C. (2008). Pickup and delivery with split loads. *Transportation Science*, 42(1):32–43.
- Oldham, P. (2016). UK Freight: in for the long haul. *Institution of Mechanical Engineers*.
- Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, 58(1):21–51.
- Pérez-Bernabeu, E., Juan, A. A., Faulin, J., and Barrios, B. B. (2015). Horizontal cooperation in road transportation: a case illustrating savings in distances and greenhouse gas emissions. *International Transactions in Operational Research*, 22(3):585–606.
- Ropke, S. and Cordeau, J.-F. (2009). Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3):267–286.
- Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472.
- Ruland, K. and Rodin, E. (1997). The pickup and delivery problem: Faces and branch-and-cut algorithm. *Computers & mathematics with applications*, 33(12):1–13.
- Savelsbergh, M. and Sol, M. (1998). Drive: Dynamic routing of independent vehicles. *Operations Research*, 46(4):474–490.
- Savelsbergh, M. W. and Sol, M. (1995). The general pickup and delivery problem. *Transportation science*, 29(1):17–29.
- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *International Conference on Principles and Practice of Constraint Programming*, pages 417–431. Springer.
- Tang, G., Ning, A., Wang, K., and Qi, X. (2009). A practical split vehicle routing problem with simultaneous pickup and delivery. In *Industrial Engineering and Engineering Management, 2009. IE&EM'09. 16th International Conference on*, pages 26–30. IEEE.
- Venkateshan, P. and Mathur, K. (2011). An efficient column-generation-based algorithm for solving a pickup-and-delivery problem. *Computers & Operations Research*, 38(12):1647–1655.
- Wang, H., Du, L., and Ma, S. (2014). Multi-objective open location-routing model with split delivery for optimized relief distribution in post-earthquake. *Transportation Research Part E: Logistics and Transportation Review*, 69:160–179.
- Wang, X. and Kopfer, H. (2014). Collaborative transportation planning of less-than-truckload freight. *OR spectrum*, 36(2):357–380.
- Xu, H., Chen, Z.-L., Rajagopal, S., and Arunapuram, S. (2003). Solving a practical pickup and delivery problem. *Transportation science*, 37(3):347–364.