Soltani, Shaghayegh, Taylor, P.S., Parker, Edward A. and Batchelor, John C. (2020) *Popup Tunable Frequency Selective Surfaces for Strain Sensing.* IEEE Sensors Journal . pp. 1-4. ISSN 1530-437X.

An Effective Real Time GRASP-Based Metaheuristic:
Application to Order Consolidation and Dynamic Selection of Transshipment Points for Time-Critical Freight Logistics

Said Salhi[*a], Brian Gutierrez[a], Niaz Wassan[a-b], Shaomin Wu[a], Rukiye Kaya[a]
[a] Centre for Logistics & Heuristic Optimization, Kent Business School,
University of Kent, Canterbury, CT2 7FS, UK
[b] Department of Operations Management and Business Statistics,
College of Economics and Political Science, Sultan Qaboos University, Oman.

Emails: s.salhi@kent.ac.uk ; Brian.Gutierrez@priorityfreight.com; N.A.Wassan@kent.ac.uk
([b] n.wassan@squ.edu.om); S.M.Wu@kent.ac.uk; rk398@kent.ac.uk

Abstract

Time-critical freight logistics is an area within logistics research where the shipper's orders need to be received relatively urgently using a third party logistics (3PL) that provides a quote (bid) to the shipper within a very short period. We solved this 3PL problem by developing an effective meta-heuristic based on the Greedy Randomized Adaptive Search Procedure (GRASP). This is achieved by introducing novel attributes in the construction of the restricted candidate list while incorporating flexible and intelligent rules, some of which are inspired by expert knowledge. The approach performs order consolidation, locates transshipment points and performs an optimal assignment of shipments to the selected consolidation points dynamically and in real time. This intelligent system embeds expert knowledge within the design of neighbourhood reduction schemes and data structures to speed up the search. This is achieved by recording computed data that does not need to be recomputed again while avoiding unnecessary computations of the non-promising alternatives.

The performance of this real time optimisation and scheduling tool is tested with a European 3PL company over a 13 weeks period in late 2017 resulting in a significant cost saving and a considerable reduction in CO_2 emissions. This powerful decision support system assists the 3PL company in gaining competitive leadership advantage through producing promising quotes that turn customer requests into real customer orders.

Keywords – Less Than Truckload operations; time critical freight logistics; GRASP; order consolidation and transshipment; CO_2 emissions.

[*]Corresponding author

1. Introduction

In the manufacturing sector as well as in many service industries, logistics and especially the transportation activity is usually outsourced to third party logistics (3PLs). Though this activity is vital to the success of most companies, it is also worth noting that it is one of the major contributors to energy consumption and greenhouse gas emissions. For instance, in the EU-28 (European Environment Agency, 2018a), CO_2 emission, due to transport but excluding international flights and shipping, has risen by over 18% from 1990 to 2016. This study will contribute to the reduction in both the operating costs (economic benefit) and CO_2 emission (environmental impact). This will be achieved by adopting an intelligent scheduling tool that incorporates the power of metaheuristic and the importance of expert knowledge which leads to using less mileage and fewer vehicles on the roads.

The problem that we investigate is related to time-critical freight logistics where the shipments (customer orders) need to be served urgently. For example, such requests may be due to machine breakdowns, which can have a detrimental effect on production in the manufacturing industry. To speed up the process, these shippers phone (or email) 3PL transport agencies specialised in expedient logistics to deliver their items very quickly. These agencies do not usually have their own vehicle fleet but rely on common carriers to perform the collection and delivery operations.

A classical event is that a shipper phones a few of these 3PLs and provides the necessary information of the shipment to the operator of these 3PL companies. For example, this includes the collection location of the requested part, its destination or delivery location, its due date (usually a tight due date), and all of its characteristics such as its weight, volume, dimensions, fragility, etc. The operators of these 3PLs need to come up with a quote (bid) within a very short period say 10-15 minutes, even though the shipment itself may take several ho

urs to reach the destination (e.g., bringing a car engine from South of Spain to the Midland in the UK). The 3PLs usually base their bid on their past information related to direct shipments using full truckload operations.

The work flow of our 3PL company is illustrated in Figure 1. Given that there are several such 3PL companies competing for the same type of service (emergency type delivery), they usually provide quotes nearly instantly to the shippers who may or may not accept their quotes. If a quote is accepted, the chosen 3PL company then organises the collection of the required part and its

delivery to the shipper by paying the chosen common carrier while still guaranteeing a reasonable profit margin. A high cost quote may lead to losing the shipment whereas a low cost quote may generate a loss as the carrier cost may be higher than expected. Providing a competitive quote within a very short time period is therefore critical to the success of the business.
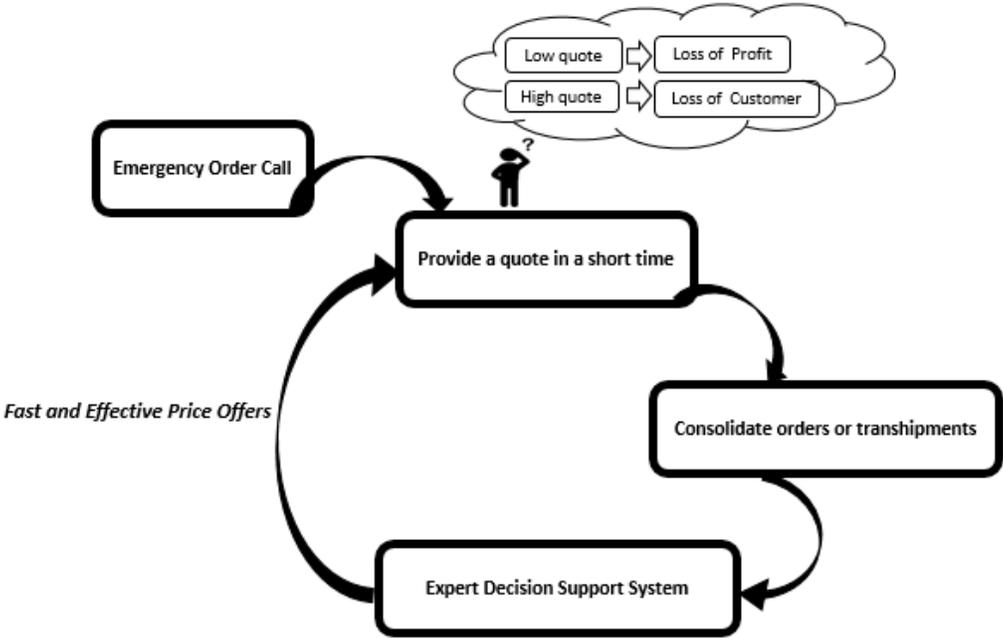
Figure 1: Work Flow of the 3PL Company

In our case study project, the 3PL company does not have its own fleet and acts as a mediatory transport agency, which attracts shipments from all over the world, and guarantees their collection and their delivery through common carriers on road operations or by air if necessary. The company employs 30-40 experienced and multilingual operators whose tasks are to respond to calls, provide quotes and identifying the right common carriers from anywhere in Europe, North America and recently China. The company's goal is to maximize its profit by securing as many shipments as possible while retaining a high level of service through the selection of suitable carriers (known as transport suppliers).

To be responsive and retain its customer service as high as possible, the 3PL company, as many others also do, tends to assign an empty truck or a van directly for each shipment (individual order)

irrespective of the shipment load which is usually much smaller than the full truckload (TL). Though this strategy of opting for TL lookalike operation sounds sensible and safe, the quote the logistic company provides may sometimes be higher than necessary to guarantee a reasonable profit margin, which unfortunately may, in many occasions, lead to a large number of lost orders.

To assist this 3PL company to resolve its day-to-day operations which is performed by hand and relying on the 30-40 operators' experience, we developed an effective real time decision support scheduling tool based on a metaheuristic, namely, GRASP. This intelligent optimisation tool takes advantage of the less than truckload (LTL) operations while incorporating the 3PL operators' expert knowledge and experience. This invaluable information helped us to design a data structure and neighbourhood reduction mechanisms that contributed significantly in speeding up the search. This innovative but easy to use tool has significance and a massive impact on the 3PL company as it enables them to embrace the new technology which resulted in cutting down cost and attracting relatively more quotes, and more profit margin. As a by-product, the tool reduces the amount of CO_2 emissions as a result of using fewer vehicles on the roads.

Our optimisation tool aims to identify very quickly the pairs of shipments that can be consolidated into one vehicle (truck, van) either en-route or through locating and assigning potential transshipment (trans-loading or stopover) points where consolidation can take place. This challenging exercise differs from most earlier studies which we will discuss next in the review section. There, the shipments are usually known a long time before hand and the shippers bid for the lanes whereas in ours this is not the case. Our objective is to minimize the total financial cost and environmental impact gained from consolidating freight while maintaining the high level of customer service. A feasible consolidation of freight from two vehicles into one saves not only on wasted unnecessary mileage, the number of empty movements and the use of under-utilised vehicles, but also reduces traffic congestion and CO_2 emission and hence contributes positively to the environment. Though this case study is limited to exploring the potential benefits of combining shipment pairs only due to maintaining customer service and company's reputation in terms of customer relationship, this study could obviously be extended, as will be highlighted in the suggestion section, to cater for triplets or more if other similar companies wish to opt for a different business strategy.

4

This problem shares some similarities with (i) the area of freight consolidation and (ii) the area of pickup and delivery with transshipments. In part (i), LTL freight are combined to make bundles that are assigned to trucks turning the problem into a clustering and a multi-commodity flow routing problem. Interesting forms of consolidation are presented and analysed in Hall (1987) and Campbell (1990). In part (ii), from a general routing perspective, this problem has also some similarities with the pickup and delivery problem with transshipment (PDPT) which we will also discuss in the review section. Information on this topic can be found in Rais et al. (2014).

This challenging logistic problem, though it has a massive practical importance, has been dormant for over two decades. It is only in the last few years where this issue has re-emerged considerably, as will be shown in the literature review section. This is mainly due to the high increase in volume of freight, leading to the outsourcing of the transportation activity to the 3PL companies which has led to the high level of competition between these companies. This issue has now become contemporary and important due to public awareness of the effect this may have on the environment which is now supported by governments and worldwide organisations. For example, the European Environment Agency (2018b) has recently reported in August 2018 that the energy consumption due to road transport has increased by nearly 32% from 1990 to 2016 in the EEA33 countries.

The contribution of this study is three fold:

(i) highlight the need for a fast but effective optimisation tool to deal with time-critical freight logistics for the case of LTL operations,

(ii) propose an efficient metaheuristic such as the Greedy Randomised Adaptive Search Procedure (GRASP) that incorporates expert knowledge and intelligent rules when consolidating shipments within trucks or through a dynamic allocation of s points (trans-loading, stopover),

(iii) demonstrate the economic benefit (reduction in the total cost) and the environmental and health impact (reduction of CO_2 emissions) of this effective decision support system in a real life logistical setting.

The remainder of the paper is organised as follows. The literature review is given in Section 2 followed by the methodology in Section 3. This includes the consolidation of orders, the transshipment points and the calculation of the cost saving. The proposed approach is described in

Section 4 and the design of speed up mechanisms are given in Section 5. The experiments using real life data from a freight company in the South of England are shown in Section 6. The conclusion and limitations of the study alongside potential research avenues are given in the last section.

2. Literature Review

Studies that deal with the benefits of combining shipments for the case of LTL operations have started to come up though their number is still relatively small compared to classical transportation or vehicle routing problems. This section aims to review some of these papers which we like to categorise under two classes, namely, (i) freight consolidation and (ii) pickup and delivery with transshipment.

(i) *Freight consolidation-* Our problem has some similarities with the combinatorial auction problem where the shipper after gathering all the bids about the freight lanes from the carriers solves the winner determination problem by matching lanes with carriers (Abrache *et al.* 2007). However, in the carrier's problem the lanes are made available beforehand and the question is to come up with attractive bids. This is usually solved as a minimum cost flow problem (Chang, 2009). These models are mainly focussed on carriers for TL operations where the shipments are sent from one origin to one destination using one truck.

For instance, Ulku (2012) presents an interesting study that demonstrates the economic and environmental benefits of shipment consolidation. The benefits of economy of scale by using fewer long haul shipments through combining two or three shipments when possible showed to result, in a large number of cases, in higher truck loading (vehicle utilisation, ton-miles per truck).

Mesa-Arango and Ukkusuri (2013) provide an easy to follow illustrative example on how to achieve economy of scale. The authors also extend the TL related work to cater for Less than TruckLoad (LTL) operations where they show that consolidation of shipments in LTL operations is likely to yield attractive bids. The idea is to determine the appropriate bundles (set of lanes) where each bundle can be served by one truck. They produce a novel mathematical formulation by turning the problem into a multi-commodity one to one pick up and delivery TSP problem.

6

They solve smaller instances (4 shipments) to optimality using a Branch and Price algorithm enhanced by clever acceleration strategies.

Estrada-Romeu and Robuste (2015) examine the impact of stopovers in LTL operations by allowing for more than one additional shipment to be inserted for possible consolidation at such stopovers or/and hubs. To speed up the search, they identify promising stopovers among existing distribution centres using novel analytical formulae. They show that only those locations that lie within the sphere whose loci are the origin and the destination of the shipment are worth examining. This reduction scheme is then successfully embedded into a tabu search metaheuristic producing interesting results.

In these earlier studies, contrarily to ours, all the new shipments are known to the carriers before hand, most carriers already have current contracts (lanes already allocated to them) and above all the decision making process is not performed in real time. In our study, the 3PL company receives phone calls or emails requesting quotes (bids) for the delivery of the shipments which belong to a large number of shippers. These shippers expect decisions to be made in a very short time (say within 10 to15 minutes) even though their shipment may take several hours to be delivered. For example it may originate from Spain or Italy to be destined for the Midland in the UK. In addition, restrictions such as time windows, the use of various type of trucks, the dynamic nature of the transshipment point selection, render the problem very special and practically challenging. In this work, we consider the most profitable consolidation of shipments which can be achieved either by merging shipments en route using one truck only or by choosing suitable transshipment point where the consolidation or trans-loading can take place leading to one of the truck to terminate its journey. Though the operation itself may take days to complete, the decision has to be made in a very short time so to assist the 3PL company in gaining the bid while ensuring a profitable margin. The chosen common carrier will then be asked to consolidate en route or to pass via a transshipment point. In case of the latter, the truck either terminates and its shipment be transferred to another truck, or continues its journey after receiving the shipment of another truck.

(ii) Pickup and delivery problem with transshipment- Our problem can also be considered as a generalised case of the pickup and delivery problem with transshipment (PDPT) with additional constraints. The introduction of transshipments add another dimension to the formulation to the well know pure PDP or PDP with time window. An interesting review on PDP is given by

Berbeglia *et al.* (2007) while several variants are described in Parragh *et al.* (2008). For more information and recent references in this area, see Al-Ghani *et al.* (2018) and Lu and Yang (2019).

The closest routing to ours is the PDPT, which is shown to be NP hard as it is a generalisation of the PDP which is NP hard (Lenstra and Kan (1981), Rais et al. (2014)). There is however a shortage of published work on PDPT, compared to its counterpart the PDP. Shang and Cuff (1996) were among the first to address this particular routing problem. For simplicity, we classify these studies under exact methods and heuristics. More information on the classification of these two methodologies in general can be found in Salhi (2017).

Exact methods- Cortes *et al.* (2010) formulate the problem as a generalisation of the PDP. Here, a request is allowed to be served not necessarily by one vehicle but through a transshipment or a transfer point also. A branch and cut approach using Bender decomposition is adopted to avoid the weakness of the LP relaxation. This approach results in optimally solving instances with six requests while speeding up the classical mixed integer programming model by nearly 90%. Masson *et al.* (2014) implement a branch and cut and price algorithm for a variant of the PDPT where a request has to be delivered by a single trip from the transshipment point without collecting other loads. This particular problem was solved optimally up to 85 requests. Rais *et al.* (2014) develop an efficient mixed integer programming formulation for the problem. Instances with 14 nodes (i.e., 7 requests) are solved to optimality while requiring a large amount of CPU time. The authors also observed that the time increases drastically with the number of nodes. For example, 10 node instances consume about one minute on average while 14 node instances require nearly 3 hrs. This sharp increase in computational burden limits the use of exact methods and therefore provides an opportunity for heuristics to fill the gap, as outlined later.

Heuristics- Shang and Cuff (1996) solve the PDPT for a health maintenance organisation by adopting a look ahead insertion heuristic. Mitrovic-Minic and Laporte (2006) develop a simple but effective two phase heuristic. In phase one, several random solutions are generated and the best one is improved in phase two, which is achieved by removing and reinserting one request at a time. Instances with up to 100 requests are randomly generated under various scenarios whose results are led to interesting managerial insights. Thangiah *et al.* (2007) extend the earlier insertion type approach by introducing a local search. Their aim is to reduce the size of vehicle fleet by removing requests from the lighter vehicles and inserting them into other ones. Qu and Bard (2012) devise a

composite heuristic made up of the Greedy Randomised Search Procedure (GRASP) followed by Large Neighbourhood Search (LNS). The former is used to obtain initial solutions while the latter is to improve them using various removal and repair mechanisms. Instances with 25 requests that are purposely constructed and for which optimal solutions can be derived are used for comparison purposes. The authors develop two insertion procedures: one for single route without transshipment and the other, known as double insertion, is for routes with transshipments. An interesting speedup mechanism using a tree based data structure is also incorporated into the search. This enhancement resulted in requiring about 200 secs for GRASP and 1200 seconds for LNS (i.e., over 20 minutes altogether). Masson *et al.* (2013) put forward an adaptive LNS to solve the PDPT for a school bus company. Their approach is tested on the 100 request instances provided by Mitrovic-Minic and Laporte (2006) with encouraging results. Recently, Danloup *et al.* (2018) revisit the PDPT by developing a Genetic Algorithm (GA) and LNS. The authors tested their two algorithms on instances with 25 and 50 requests from the literature. An efficient and fast evaluation mechanism of the double insertion heuristic of Qu and Bard (2012) is also developed. This intelligent feature speeded up the search considerably, which leads their methods to produce competitive results with CPU times varying from 2 minutes to approximately 15 minutes.

Our problem also resembles, in a small way, the ride-sharing problem which is an extension of the dial a ride problem where people, instead of load, are sharing the same vehicle. Interesting studies on this topic include Agatz *et al.,* (2011) and Furuhara *et al.,* (2013).

Though our problem has some similarities with some of the studies mentioned above, our problem needs to

(i)   be solved in real time (i.e., quick response time to the shipper) and

(ii)   incorporate, for a given pair of shipments, the dynamic selection of the transshipment with appropriate characteristics and attributes (i.e., material handling, opening hours, safety standard, cost, …)

(iii) satisfy the following real life constraints

- the multi-dimensional capacity constraints (i.e., weight, height, length and width),

- the compatibility of the products to be consolidated (fragility,…),

- the heterogeneity of the vehicles,

-   the possibility of merging the two shipments (or splitting the total load of the two shipments into two distinct ones) en route or through transshipment points,

-   the restriction of the trucks not to necessarily return to the same depots,

-   other restrictions that may include company preferences and guidelines to retain high customer service, shipper's requirements due to confidentiality, among others.

The richness of this logistic problem makes its resolution by exact methods inappropriate as it is a generalisation of PDPT which is already noted to be NP hard earlier in this section. The only way forward is therefore to adopt an effective scheduling system that combines expert knowledge and the power of metaheuristics. The added advantage of this tool is that, besides being fast and implemented in real time, it has the flexibility to accommodate new information from the expert at any time while still providing a good solution. For more details on heuristic search in general, the reader will find the recent edited book by Gendreau and Potvin (2019) and the book by Salhi (2017) to be informative. In this study, we adopt the metaheuristic GRASP which will be discussed in Subsection 4.3.

3. Methodology

In this section, we present an overview of our algorithm, the notations used and the configurations that need to be identified and evaluated a priory. The overall cost saving including the corresponding cost saving for every consolidated pair of shipments is also determined. Note that the problem can be formulated as a mixed integer linear programming (ILP) problem by introducing a lot of modifications to the MPPDP mathematical formulation given by Rais et al. (2014). For very small instances, this may be solved optimally using commercial optimisation software such as ILOG CPLEX or Gurobi. However, for the case of larger instances metaheuristics could be attempted. Due to the restriction of using only FTL or consolidation of pairs only through en-route or via transshipment points, instead of providing a more generalised ILP formulation, we introduce optimal configurations of pairs of shipments that are used in a simpler ILP model or in a meta-heuristic. This approach is shown to be promising for larger instances as in this case study and which need a solution in real time.

3.1 Algorithm Overview

We propose a three-phase approach that makes up our decision support system. Phase 1 comprises the necessary input as well as the construction and updating of the data structure which stores the information about each shipment. Phase 2 consists of determining the best cost saving for consolidating each potential pair of shipments. Here, we also identify those shipments that are worth consolidating and how the consolidation is taken place whether it is through en route or via a transshipment point (trans-loading, stopover). This phase forms the main bone of the approach, which is solved using the meta-heuristic GRASP whose description is provided in Subsection 4.3. Phase 3 displays the suggested consolidated shipments which assist the operators in providing a competitive quote. This phase also controls the time update including the input of the new orders which are then fed back to phase 1. The overall process runs continuously and in real time with a regular time update at every Δ time. In our experiments, as requested by the 3PL company, Δ is set to one minute. This is mainly due to the high volume of shipment requests, currently controlled by 30-40 operators. The three phases of the algorithm are briefly outlined in Figure 2.

Phase 1: At time t, input (update) all the shipments, and construct (update) the data
structure that contains all the information for consolidation for each shipment,
including the set of potential transshipments (see Subsection 5.1).

Phase 2: Determine the best configuration using the GRASP metaheuristic
(see Subsection 4.3)

Phase 3: Set $t = t + \Delta$. Display the results, update the information about the shipments
including the new ones and go to Phase 1.

Figure 2: The dynamic setting of the three phases of the decision support system

3.2 Notation

Parameters and Index Sets:

Δ: time update of the running of the algorithm (in our experiments, Δ is one minute)

n : number of shipments, indexed by $i \, (i = 1,...,n)$

C_i: the collection location of shipment $i \, (i = 1,...,n)$

D_i: the destination location (delivery) of shipment $i \, (i = 1,...,n)$

11

Volume(*i*), *Weight*(*i*), *Fragility*(*i*), *Height*(*i*), *Width*(*i*), *Length*(*i*) :

    These are the basic characteristics of transshipment i $(i = 1,...,n)$

$[A_i, B_i]$: time window for transshipment i with A_i and B_i being the earliest and latest delivery time

        for transshipment i $(i = 1,...,n)$

$\mathbf{V} = \{(C_i, D_i); i = 1,...,n\}$: set of shipments $(|\mathbf{V}| = n$), each shipment i being defined by its

      collection and delivery locations C_i and D_i; $i = 1,...,n$ respectively

$\Re$: set of regions, indexed by R_r; $R_r \in \Re$; $r = 1, ... , |\Re|$

$\overline{R}_r$: the base location (reference point, say city centre location of the main city) defining region

    R_r; $R_r \in \Re$; $r = 1, ... , |\Re|$

$d_{P_i P_j}$: distance (cost) between point locations P_i and P_j where a point location refers to

    a collection (origin), a delivery (destination) of shipments i, j $(i, j = 1,...,n)$, or simply the

    location of a transshipment point, or the base location of a given region

$\mathbf{P}$: set of potential transshipment points (stopovers, trans-loading point) throughout the

    business area $\Re$, indexed by transshipment point T_k ; $T_k \in \mathbf{P}$ $(k = 1,...,|\mathbf{P}|)$

$\mathbf{A}$: set of attributes for the transshipment points with $\mathbf{A} = \{$opening hours, handling equipment,

    reliability, financial charge$\}$

$\mathbf{a}_s^k$: the sth attribute of the kth transshipment point $T_k \in \mathbf{P}$ $(k = 1,...,|\mathbf{P}|; s = 1,...,|A|)$)

$\mathbf{K}$: set of potential configurations of serving any two shipments $(\mathbf{K} = \{0,1.,2,....,8\})$, indexed

    by $v \in \mathbf{K}$ with $v = 0$ referring to the original configuration (no consolidation)

    $v \in \{1,...,4\}$ for en-route consolidation and $v \in \{5,...,8\}$ for transshipment

    consolidation

$\mathbf{\Omega}_{rs}^l$: subset of potential transshipment points $(\mathbf{\Omega}_{rs}^l \subset \mathbf{P}$) that can be used between shipments

    with collection (delivery) points in regions R_r and R_s and the delivery (collection) point in

    region R_l; $R_r, R_s, R_l \in \Re$

Decision Variables:

C_{ij}^1 : the least cost of consolidating shipment i with shipment j; $i, j = 1, ..., n$

C_{ij}^2 : the second least cost of consolidating shipment i with shipment j $(i, j = 1, ..., n)$

S_{ij} : cost saving over the original configuration when using the best consolidation

configuration of shipment i with shipment j $(i, j = 1, ..., n)$

$$X_{ij} = \begin{cases} 1 & \text{if shipments } i \text{ and } j \text{ are consolidated; } i, j = 1, ..., n \\ 0 & \text{otherwise} \end{cases}$$

$\mathbf{N_i}$: subset of shipments that is worth exploring for possible consolidation with shipment

i $(i = 1, ..., n)$ with $\mathbf{N_i} \subset \mathbf{V}$ and $|\mathbf{N_i}| << |\mathbf{V}|$, see Subsection 5.2.

π_{ij}^v : cost of serving shipments i and j using configuration $v \in \mathbf{K}$, see Subsection 3.3 for

$v \in \{1, ..., 4\}$ and Subsection 3.5 for $v \in \{5, ..., 8\}$

Γ_{rs}^l : subset of non-dominated transshipment points ($\Gamma_{rs}^l \subseteq \Omega_{rs}^l$) that can be used between

collection (delivery) points in regions R_r and R_s to go to (come from) region R_l; $R_r, R_s, R_l \in \mathfrak{R}$,

see Subsection 3.4.

3.3 Generation of the Configurations for Consolidation

The consolidation is achieved through either (a) order consolidation or (b) transshipment consolidation whichever option leads to the least expensive feasible configuration.

We first describe the feasibility conditions for consolidating shipment i with shipment j; $(i, j = 1, ..., n)$.

Feasibility conditions-

We consider the consolidation of shipments i and j to be feasible if

1- $Volume(i) + Volume(j) \leq TruckVolume$

2- $Weight(i) + Weight(j) \leq TruckWeight$

3- $Max[height(i), height(j)] \leq TruckHeight$

4- $Max[length(i), length(j)] \leq TruckLength$

5- $Max[width(i), width(j)] \leq TruckWidth$

6- Time windows for both shipments i and j, namely, $[A_i, B_i]$ and $[A_j, B_j]$, remain satisfied if the two shipments are consolidated

7- Both shipments i and j are not restricted by their respective shippers to be performed individually.

Note that the chosen truck to be used is the one with characteristics, namely, *TruckVolume, TruckWeights, TruckHeight, TruckLenght* and *TruckWidth*, and which is the cheapest that can accommodate both shipments.

(a) Scenario 1- En route Order Consolidation Configurations

Consider two shipments i and j whose collection points are C_i and C_j with their delivery destinations being D_i and D_j respectively.

The traditionally adopted schedule by the company is to adopt a TL operation strategy by performing the two trips, one for each shipment. This requires two trucks

$C_i \rightarrow D_i$ and $C_j \rightarrow D_j$, resulting in a cost of $\pi_{ij}^0 = d_{C_i D_i} + d_{C_j D_j}$ (1)

In this scenario, four possible configurations need to be evaluated as long as feasibility constraints are not violated. These are summarised in Table 1.

Table 1: Configuration details of order consolidation (case of Scenario 1)

#	Configuration	Cost	Equation #
[1]	$C_i \rightarrow C_j \rightarrow D_i \rightarrow D_j$	$\pi_{ij}^1 = d_{C_i C_j} + d_{C_j D_i} + d_{D_i D_j}$	(2a)
[2]	$C_i \rightarrow C_j \rightarrow D_j \rightarrow D_i$	$\pi_{ij}^2 = d_{C_i C_j} + d_{C_j D_j} + d_{D_j D_i}$	(2b)
[3]	$C_j \rightarrow C_i \rightarrow D_j \rightarrow D_i$	$\pi_{ij}^3 = d_{C_j C_i} + d_{C_i D_j} + d_{D_j D_i}$	(2c)
[4]	$C_j \rightarrow C_i \rightarrow D_i \rightarrow D_j$	$\pi_{ij}^4 = d_{C_j C_i} + d_{C_i D_i} + d_{D_i D_j}$	(2d)

To eliminate those configurations that are unlikely to lead to a promising outcome, neighbourhood schemes are also introduced in Subsection 5.2.

(b) Scenario 2- Transshipment Consolidation Configurations

In many cases, consolidation opportunities with one vehicle throughout an entire route may be missed due to the widespread distances between the collection points, even though the delivery points may be relatively close. A similar but symmetrical case is when the collection points are close to each other but the delivery points are far apart. This latter case will be considered afterward to retain clarity.

Case 1: Merging Deliveries

As an example, consider a job collecting in Barcelona (Spain) and delivering to Birmingham (UK) area, and another collection in Hamburg (Germany) for delivery to the Birmingham area. These two shipments will not be considered for order consolidation as described in Scenario 1 due to the collection points being too far apart. However, if a network of transshipment points throughout Europe is established, both shipments may be considered for consolidation through one of the transshipment points if found profitable and practically possible (e.g., feasibility conditions satisfied). In other words, the two collections are performed in two separate trucks which are then consolidated at the most suitable transshipment point into one truck which will then continue serving both delivery points to the Birmingham area while the other truck will not be used.

In general terms, this merging deliveries scenario is based on having two shipments with origins C_i and C_j, and destinations D_i and D_j respectively. These shipments can be merged at the transshipment point T_k leading to two further configurations, namely [5] and [6], as described in Figure 3. For clarity of presentation, their respective costs will be defined in Subsection 3.5.

Figure 4 shows two shipments with distant collection points, region R_r (Barcelona area in Spain) and region R_s (Rome area in Italy), both with proximate deliveries in region R_l (Berlin area in North Germany). Here, several potential transshipment points can be identified. These are widely spread starting in the Southeast of France, followed by the North of Italy, South of Germany and so on.
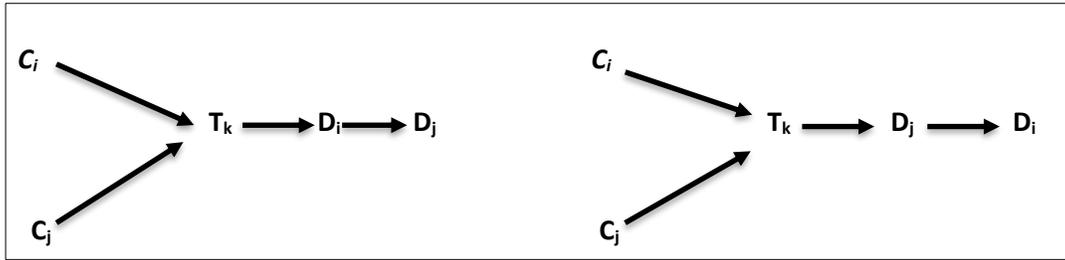
Figure 3: Merging deliveries leading to two configurations [5] (left) and [6] (right)
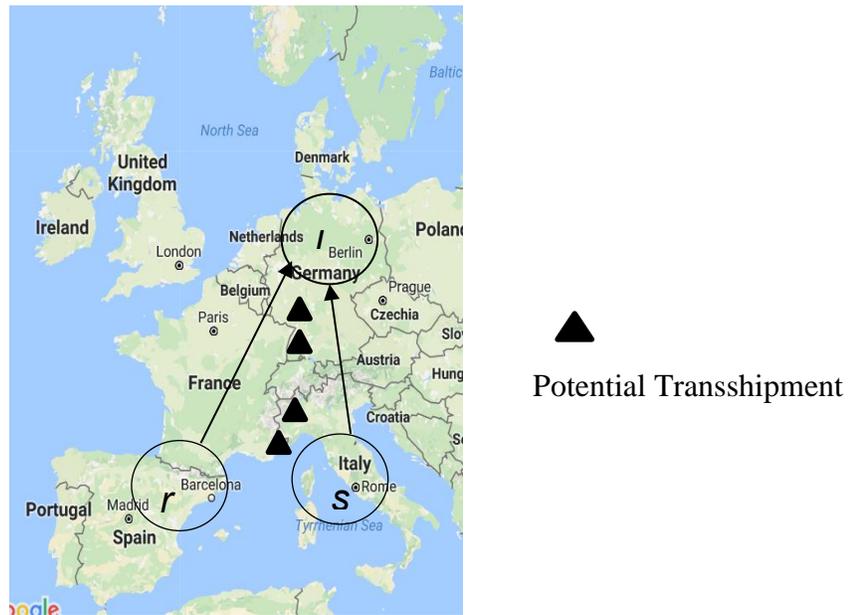
Figure 4: Example of Transshipment Points in Scenario 2 (Case 1)

Impact of neighbourhood reduction- Based on the three regions $\{R_r R_s, R_l\}$, a subset of these transshipment locations will be chosen as the non-dominated transshipments points (see Subsection 3.4), from which the optimal transshipment point for these two shipments will be selected. This reduction mechanism will result in an important reduction in computational time by avoiding evaluating unnecessary transshipment points.

Vehicle or truck size- If both shipments are still in the planning stage, a vehicle large enough to consolidate both shipments is chosen at the collection point closest to the selected transshipment point. However, if one of the two shipments has already been planned for a given vehicle, and can feasibly accommodate both shipments, this vehicle would be used for the consolidation at the transshipment point irrespective of whether or not it originates from the nearest collection point to

this transshipment point. This strategy is adopted to retain the strong relationship between the 3PL company and its transport suppliers (common carriers).

Case 2: Merging Collections

This consolidation case, though similar in configuration to the merging deliveries described in the first case, has two shipments that have proximately close collection points but distant delivery points. Both collections are first performed by one large enough truck up to the transshipment point. The freight is then split into two separate trucks to deliver to their respective delivery points. In this case, the vehicle used to collect both jobs continues from the transshipment point to its delivery point while a second vehicle begins its journey at the transshipment point and delivers to the other customer. The first truck, as it may be the biggest, will choose the shortest journey between the transshipment point and the two delivery points. In other words, as in scenario 1, we aim to use the smallest vehicle if necessary for the longer remaining journey.

In this case, the objective is to consolidate at the point *farthest* from the collection region, contrarily to the first scenario where the near transshipments to the collection points are usually profitable. This scenario has two different collection configurations ([7] and [8]), as seen in Figure 5. For clarity of presentation, their respective costs will also be defined in Subsection 3.5.
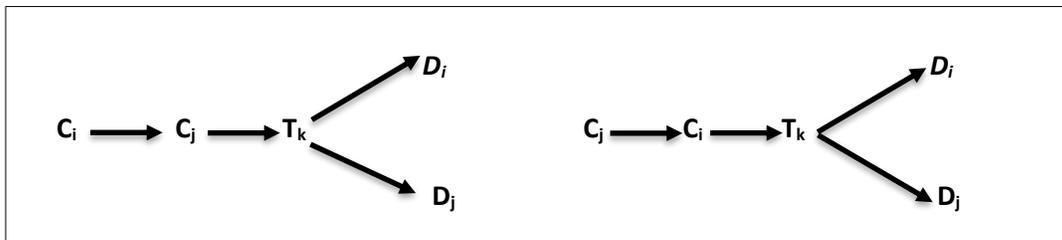
Figure 5: Merging collections configurations [7] (left) and [8] (right)

3.4 Determining the Set of Non-Dominated Transshipment Points (Case 1)

As an example, consider Figure 4 again. The ideal transshipment point would be the location in the Southeast of France (SF). This would produce the largest distance savings as one vehicle is kept along the route as long as possible. Next, the point, in North of Italy (NI), will be assessed against the point in SF. If there is a certain attribute that NI can fulfil which SF cannot, thus, NI is

17

not dominated. It may, for example, be open on the weekends for longer hours than SF. NI will then be part of the final set alongside SF. Next, another point in the South of Germany (SG) will be assessed against the non-dominated points, namely, SF and NI. If every attribute of SG can be fulfilled by either NI or SF, then SG will be dominated and will not be included in the final set and hence discarded from subsequent investigation. This identification mechanism continues until we are left with the set of non-dominated transshipment points only.

Each potential transshipment point is defined by the following four attributes in the set $\mathbf{A}$.

Attribute 1- the opening hours (say 8am-5pm, 8am till midnight, or 24/7),

Attribute 2- the type of handling equipment (heavy, light, simple task, multiple tasks, fast, slow),

Attribute 3- reliability (how much reliable at loading or unloading)

Attribute 4- the financial charge (fixed, time dependent or load dependent or both)

To assign the most profitable transshipment point for a given pair of shipments, we just need to choose among the non-dominated transshipment points only. This reduction mechanism avoids evaluating unnecessary combinations which is paramount in our study given the problem is solved in real time.

Let us formally define this set of non-dominated points. First, the set of potential transshipment points between a triplet of regions $\{R_r, R_s, R_l\}$ is identified in (3) as follows

$$\mathbf{\Omega}^l_{rs} = \{T_k \in \mathbf{P} \mid C_i \in R_r, C_j \in R_s, D_i \in R_l, D_j \in R_l \text{ or } C_i \in R_l, C_j \in R_l, D_i \in R_r, D_j \in R_s\} \quad (3)$$

Note that (3) defines the transshipments that are close to the shipments which originate from the two regions $\{R_r, R_s\}$ but their destination happens to be in the same region R_l. An example is shown in Figure 4. Equation (3) also defines the opposite case, namely, case 2 as displayed in Figure 5. Here, the transshipment points are closer to the collection points which belong to the same region but their corresponding delivery points are not.

For convenience but without loss of generality, we explain the main steps based on Case 1 and then modify them accordingly, and whenever necessary, for Case 2.

For each pair of shipments $\{(C_i - D_i), (C_j - D_j)\}$ that originates from R_r and R_s, and whose destination happens to be in the same region R_l, the corresponding subset of the non-dominated transshipment points in $\boldsymbol{\Omega}_{rs}^l$, which we denote by $\boldsymbol{\Gamma}_{rs}^l \subset \boldsymbol{\Omega}_{rs}^l$, is constructed using the following scheme:

The non-dominance based scheme

(a) Order all transshipment points in $\boldsymbol{\Omega}_{rs}^l$ starting from the nearest to R_r and R_s using the base location of each region as a reference point, say $(\bar{R}_r, \bar{R}_s)$. For instance, the first one is defined as $\gamma(T_1) = \text{Arg} \underset{T_k \in \Omega_{rs}^l}{\text{Min}} (d_{\bar{R}_r T_k} + d_{\bar{R}_s T_k})$. This order relationship leads to the following

$$\gamma(T_1) \prec \gamma(T_2) \prec ... \prec \gamma(T_{|\Omega_{rs}^l|}) \text{ with } \gamma(T_1) \text{ being the nearest to } (\bar{R}_r, \bar{R}_s) \text{ and so forth.}$$

In mathematical terms, this is generalised as follows

$$d_{\bar{R}_r \gamma(T_1)} + d_{\bar{R}_s \gamma(T_1)} \leq d_{\bar{R}_r \gamma(T_2)} + d_{\bar{R}_s \gamma(T_2)} \leq ... \leq d_{\bar{R}_r \gamma(T_{|\Omega_{rs}^l|})} + d_{\bar{R}_s \gamma(T_{|\Omega_{rs}^l|})} \qquad (4a)$$

(b) $\gamma(T_k)$ is considered as not dominated (known as extreme or Pareto point in the literature) if $\gamma(T_k)$ has at least one attribute whose specification is better than its preceding transshipment points in the ordered list (i.e., $\gamma(T_{k'})$; $k' < k$).

(c) Let such a subset be defined as $\boldsymbol{\Gamma}_{rs}^l = \{\gamma(T_1),....,\gamma(T_m)\} \subseteq \boldsymbol{\Omega}_{rs}^l$ with $m = |\boldsymbol{\Gamma}_{rs}^l| << |\boldsymbol{\Omega}_{rs}^l|$.

The determination of the set of non-dominated transshipment points is summarised in Figure 6.

Illustrative example for items (b) & (c) of the non-dominance based scheme

For example, $\gamma(T_k)$ has a competitive advantage by being cheaper (handling cost, etc), may have special or more reliable equipment, or it is opens more hours than the others. For an illustration, consider the following small ordered list of six transshipment points.

$$\boldsymbol{\Omega}_{rs}^l = \{\gamma(T_1),...,\gamma(T_6)\}$$

Figure 6: Determination of the set of non-dominated transshipment points (Case 1)

Assume $\gamma(T_2)$'s opening hour is wider though it is a bit more expensive than $\gamma(T_1)$, therefore $\gamma(T_2)$ is not dominated by $\gamma(T_1)$ and remains in the list. However, $\gamma(T_3)$ has similar characteristics in all four attributes than $\gamma(T_1)$ and therefore is dominated and has to be discarded. $\gamma(T_4)$'s opening hours are similar to $\gamma(T_2)$ but cheaper so it is not dominated, whereas $\gamma(T_5)$ which is like $\gamma(T_4)$ but more expensive is discarded. On the other hand, $\gamma(T_6)$ happens to be relatively more expensive than the preceding ones, but it has better equipment than the others and its opening hours is also 24/7 so it cannot be dominated.

In summary, our list reduces to $\mathbf{\Gamma}_{rs}^l = \{\gamma(T_1), \gamma(T_2), \gamma(T_4), \gamma(T_6)\} \subseteq \mathbf{\Omega}_{rs}^l = \{\gamma(T_1),....,\gamma(T_6)\}$, from which the selection of choosing the most appropriate transshipment point will take place. This is used for a given pair of shipments whose origins are located in R_r and R_s respectively while their respective destinations happen to be both in R_l.

Note that these calculations are performed from the outset and once only. These non-dominated transshipment points are irrespective of the orders requirements though these depend on the triplet of the regions $\{R_r, R_s, R_l\}$ only. It is also worth mentioning that this does not mean that all these transshipment points are feasible for any pair of shipments as some may for example violate the opening hours for a given shipment. Having said that, there will always be at least one feasible

transshipment point as the chosen subset will not have excluded any transshipment point that has extra flexibility than the others.

For Case 2, the main steps of Figure 6 remain the same except in Step 2(ii) the following changes of item (a) in the non dominance scheme are required,

    (i)   the potential transshipments are ordered so that the first transshipment point is the one that is farthest from $\bar{R}_r, \bar{R}_s$ instead (nearest to $\bar{R}_l$) with the following order relationship

$$\gamma(T_1) \prec \gamma(T_2) \prec ... \prec \gamma(T_{\Omega^l_{rs}}) \quad \text{with } \gamma(T_1) \text{ being the farthest to } \bar{R}_r, \bar{R}_s.$$

    (ii)   Step 2b is modified using the following inequality (4b) instead of (4a).

$$d_{\bar{R}_r \gamma(T_1)} + d_{\bar{R}_s \gamma(T_1)} \geq d_{\bar{R}_r \gamma(T_2)} + d_{\bar{R}_s \gamma(T_2)} \geq ... \geq d_{\bar{R}_r \gamma(T_{|\Omega^l_{rs}|})} + d_{\bar{R}_s \gamma(T_{|\Omega^l_{rs}|})} \quad (4b)$$

3.5 Computation of the Cost Consolidation in Scenario 2

Given that each shipment is defined by its collection and delivery location, we can then identify its respective collection and delivery regions $\{R_r, R_l\}$. Now consider two shipments with their respective regions. If their delivery happen to be in the same region (or their collection region is the same), we can define the set of non-dominated transshipment points as described in Figure 6. If this is not the case, these two shipments cannot be consolidated. The process continues to identify all those pairs that can be consolidated. This is achieved by determining the least cost transshipment point for any two shipments using the following order consolidation mechanism described in Figure 7.

Let $F_{\gamma(T_{k*})}$ be the financial cost incurred to consolidate the two shipments at the transshipment point $\gamma(T_{k*})$. In our case we set $F_{\gamma(T_{k*})} = a^4_{\gamma(T_{k*})}$.

For Case 1, as shown in Figure 3, there are two delivery options for the vehicle to either deliver shipment i at D_i first or serve shipment j at D_j first. The calculations of the cost for the configurations [5] and [6] (see Figure 3) are given in (5a) and (5b), respectively.

$$\pi_{ij}^5 = \underset{\gamma(T_k)\in\Gamma_{rs}^{r'}}{Min}([d_{C_i\gamma(T_k)}+d_{C_j\gamma(T_k)}+F_{\gamma(T_k)}]+[d_{\gamma(T_k)D_i}+d_{D_iD_j}]\,|\,C_i\in R_r, C_j\in R_s, D_i\in R_{r'}, D_j\in R_{r'}).\quad(5a)$$

$$\pi_{ij}^6 = \underset{\gamma(T_k)\in\Gamma_{rs}^{r'}}{Min}([d_{C_i\gamma(T_k)}+d_{C_j\gamma(T_k)}+F_{\gamma(T_k)}]+[d_{\gamma(T_k)D_j}+d_{D_jD_i}]\,|\,C_i\in R_r, C_j\in R_s, D_i\in R_{r'}, D_j\in R_{r'}).\quad(5b)$$

Step 1: If feasibility conditions for consolidating shipments i and j are satisfied (see Subsection 3.3) then locate the regions of their collection points R_r, R_s and their delivery points $R_{r'}, R_{s'}$, otherwise stop.

Step 2: If $R_{r'} = R_{s'}$ (case 1) or $R_r = R_s$ (case 2), define $\Gamma_{rs}^{r'}$ or $\Gamma_{r's'}^r$ (see Figure 6), else stop

Step 3: - Identify the non-dominated transshipment points that satisfy both shipments i and j in
  $\Gamma_{rs}^{r'}$ (see 5a-5b for case 1) or in $\Gamma_{r's'}^r$ (see 5c-5d for case 2). Let this be $\gamma(T_{k*})$.
  - Determine the least cost of consolidating i and j through $\gamma(T_{k*})$ using (5a-5b) in case 1, or (5c-5d) in case 2.
  - Record the corresponding best transshipment consolidation configuration.

Figure 7. The consolidation transshipment mechanism of shipments $i, j\,(i, j = 1,...,n)$
(case of scenario 2)

Similarly for Case 2, the cost for configurations [7] and [8], as shown in Figure 5, is defined in (5c) and (5d), respectively. For each transshipment point, there are two options for collecting either from C_i or C_j first.

$$\pi_{ij}^7 = \underset{\gamma(T_k)\in\Gamma_{rs}^{r'}}{Min}([d_{\gamma(T_k)D_i}+d_{\gamma(T_k)D_j}+F_{\gamma(T_k)}]+[d_{C_iC_j}+d_{C_j\gamma(T_k)}]\,|\,C_i\in R_{r'}, C_j\in R_{r'}, D_i\in R_r, D_j\in R_s)\quad(5c)$$

$$\pi_{ij}^8 = \underset{\gamma(T_k)\in\Gamma_{rs}^{r'}}{Min}([d_{\gamma(T_k)D_i}+d_{\gamma(T_k)D_j}+F_{\gamma(T_k)}]+[d_{C_jC_i}+d_{C_i\gamma(T_k)}]\,|\,C_i\in R_{r'}, C_j\in R_{r'}, D_i\in R_r, D_j\in R_s)\quad(5d)$$

Note that though (5a) and (5b), similarly (5c) and (5d), can be easily mathematically condensed into two equations instead of four, we have opted for this to easily identify the corresponding configuration (0 to 8) when assisting the operator of the 3PL company in his/her decision.

3.6 Computation of the Cost Saving

We first define the least cost of consolidation as follows:

The least cost of serving the two shipments i and j, including the case of non-consolidation, is computed in (6) as follows

$$C_{ij}^1 = \begin{cases} \underset{v=0,\ldots,8}{Min} \; \pi_{ij}^v & \text{if shipments } i \text{ and } j \text{ are feasible to consolidate} \\ \pi_{ij}^0 & \text{otherwise} \end{cases} \tag{6}$$

Note that the case of $v=0$ is also added in the first part of (6) as the two shipments may be feasible to consolidate though the configuration without consolidation (i.e., $v=0$) could still be the preferred option, otherwise the cost will not be defined in this special case.

The corresponding best configuration is defined by

$$v^* = Arg \; \underset{v=0,\ldots,8}{Min} \; \pi_{ij}^v \tag{6a}$$

The second least cost of consolidating the two shipments i and j is given by (7). This is similar to (6) but without the best configuration v^*.

$$C_{ij}^2 = \begin{cases} \underset{v \in \{0,\ldots,8\}/\{v^*\}}{Min} \; \pi_{ij}^v & \text{if shipments } i \text{ and } j \text{ are feasible to consolidate} \\ \pi_{ij}^0 & \text{otherwise} \end{cases} \tag{7}$$

The corresponding second best configuration is defined similarly by

$$v^+ = Arg \; \underset{v \in \{0,\ldots,8\}/\{v^*\}}{Min} \; \pi_{ij}^v \tag{7a}$$

With reference to the original configuration with cost defined in (1), and the other possible best configurations due to order consolidation costs (2a-2d) and transshipment consolidation costs (5a-5d), the overall best consolidation is derived using the following cost saving in (8)

$$S_{ij} = \pi_{ij}^0 - C_{ij}^1 \; ; \; i, j = 1,\ldots,n \tag{8}$$

The total saving is then computed in (9) as

$$TS = \sum_{(i,j) \in E^t} S_{ij} \tag{9}$$

with E^t representing those pairs of shipments that are chosen in the final solution configuration at execution period time t through GRASP, the metaheuristic that we shall present in the next section.

4 The Proposed Approach

The problem is to identify the pairs of shipments that can be consolidated by maximising the total cost saving while guaranteeing a high level of customer service. To achieve this outcome in real time, for any two shipments i and j, and at a given time period $t + \Delta$, the cost saving is computed using (8).

We first provide a mathematical formulation followed by two greedy methods and the proposed GRASP metaheuristic.

4.1 Formulation

As mentioned earlier the problem can be formulated as an extension of PDPT but due to having consolidations using pairs of shipments only and given the transformations and the cost formula that we derived in the previous section, the problem can be reduced to a simpler a 0-1 integer linear program (0-1 ILP) as follows.

$$\text{Let } X_{ij} = \begin{cases} 1 & \text{if shipments } i \text{ and } j \text{ are consolidated} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Maximise } TS = \sum_{j=1}^{n} \sum_{i=1}^{n} S_{ij} X_{ij} \tag{10}$$

Subject to

$$\sum_{j=1}^{n} X_{ij} = 1; \ i = 1,....,n \tag{11}$$

$$\sum_{i=1}^{n} X_{ij} = 1; \ j = 1,....,n \tag{12}$$

$$X_{ij} = X_{ji} ; i, j = 1,...,n \tag{13}$$

$$X_{ij} \in \{0,1\} \tag{14}$$

Objective function (10) refers to the maximisation of the total saving due to consolidation. Constraints (11) guarantee that each shipment i can consolidate with another shipment including

24

itself only and (12) show that each shipment j can be consolidated with another shipment only. Constraints (13) impose that the consolidation is unique between two shipments and (14) refer to the binary nature of the decision variables.

This ILP model has $n^2 + 2n$ constraints and n^2 binary variables only. This model is considerably smaller than the more general ILP models in the literature. This can be solved optimally using a commercial optimisation software such as ILOG CPLEX, LINDO, Gurobi or Xpress-MP for relatively larger instances than those mentioned in the literature review. Note that the value of the objective function TS is halved due to double counting (consolidating shipments i and j is the same as consolidating j and i).

Though this formulation is simple, in this occasion, it is still not practical to incorporate it in real time within the company IT system that runs every minute. However, its offline implementation on relatively small instances is useful as it acts as a benchmark for assessing the performance of the proposed metaheuristic.

4.2 Greedy-Based Heuristics

1- Basic Greedy Procedure

Step 1: Set the total cost saving $TS = 0$, time to $t = tstart$ and initialise the current set of chosen
pairs of shipments to $E^t = \{\}$

Step 2: - Select the feasible pair of shipments (i_*, j_*) as follows
$$(i_*, j_*) = Arg \max_{(i,j) \notin E^t \times E^t} S_{ij} \qquad (15a)$$
- Set $TS = TS + S_{i_* j_*}$ and $E^t = E^t \cup \{(i_*, j_*)\}$
- Remove all combinations with already chosen shipments in E^t
- If there are still feasible pairs of shipments to consolidate ((see Subsection 3.3),
repeat *Step 2*
Otherwise record all the chosen consolidated pairs in E^t and stop.

This descent type procedure could get easily trapped at a poor local maximum leading to a poor quality consolidation configuration.

2- Regret-Based Procedure

One attempt to overcome the limitation mentioned above is to adopt a look ahead strategy by incorporating the opportunity cost (also known in the literature as the regret cost). For instance, this is used in the Vogel's Approximation method to generate a good initial feasible solution for the classical transportation problem (Taha, 2017). This approach is similar to the greedy method except that in *Step 2*, instead of selecting the feasible pair that produces the largest cost saving S_{ij} using (15a), we choose the feasible pair of shipments (i_*, j_*) that yields the highest regret cost using (15b) as defined below

$$(i_*, j_*) = Arg \underset{(i,j)\notin E^t x E^t}{Max} RG_{ij} \text{ with } RG_{ij} = \hat{C}_{ij}^2 - \hat{C}_{ij}^1 ; \ \forall i, j = 1,...,n \tag{15b}$$

4.3 The Proposed GRASP-Based Metaheuristic

Both greedy-based approaches are fast but limited in scope due to their level of greediness though the regret-based method is slightly better than the basic one. These are constructive type heuristics, also known as greedy/descent-type methods, which tend to get stuck in local optima (i.e., local maxima in our case). One way to get out of such a trap is to adopt metaheuristics instead as these have the power and flexibility to escape from such local maxima. This is achieved either by allowing inferior solutions to be accepted, shifting between feasible and infeasible solution spaces, systematically extending/varying the neighbourhood of the search, destroying and rebuilding the solution in a controlled manner, or simply incorporating randomness at the construction phase of the search followed by some form of local search. In this study, we use the latter option as it is a simple adaptation of the above two greedy methods we described earlier in Subsection 4.2 besides being relatively faster than most of the others. For an overview on heuristic search in general and its applications, see the authored book by Salhi (2017) and the updated edited book by Gendreau and Potvin (2019).

4.3.1   Overview and Algorithm

GRASP metaheuristic, short for Greedy Randomised Adaptive Search Procedure, is a multi-start type approach that combines greediness with randomness. This was originally proposed by Feo and Resende (1989) but was formally presented a few years later by Feo and Resende (1995). In

brief, GRASP consists of two stages where Stage 1 refers to the construction of the solution, followed by the local search in Stage 2. In Stage 1, the solution is constructed by adding randomly one element to the partial solution one at a time from a list of promising elements. This list is known as the restricted candidate list (RCL for short). More details can be found in Resende and Ribiero (2016), Salhi (2017) with an updated review chapter in Resende and Ribiero (2019).

In this study, we also introduce two additional features to the classical GRASP implementation.

(a) The pair of shipments for consolidation is not necessarily based on one evaluation measure such as (i) the largest saving or (ii) the largest regret as commonly used in the literature. Here, one of the two measures will be randomly selected at each iteration within each run.
(b) Also, the size of the RCL is made flexible at each run instead of being fixed from the outset as usually adopted in the literature.

These two new features provide extra flexibility to the search to exploit a wider search space. RCL is made up of the promising pairs, say the ones that lie within a certain percent from the best. For instance, at each run we choose randomly $\theta \in (0,1)$ and select the ones that are within $(1+\theta)\%$ from the best. One pair is then chosen pseudo randomly from the RCL. Note that this includes the chosen consolidated pair, which can be obtained through en-route order consolidation or via transshipments. This selection is repeated until a complete solution is obtained where a local search is then activated. Given the random elements within the search, the entire process is then repeated *TMAX* times and the overall best (or average) solution is chosen.

As the problem is solved in real time and the order requests keep arriving continuously, any shipment that is practically assigned to a common carrier is then removed from the list and the updating procedure starts again. The main steps of this real time application of GRASP-based approach is given in Figure 8.

Note that in our case, the two stages of the classical implementation of GRASP are considered under Step 2 and Step 3 respectively.

Step 0 (Initialisation):
- Set the total cost saving $TS = 0$, the best saving $TS_{best} = 0$, initialise the current set of consolidated shipments to $E^t = \{\}$ and the best configuration to $E^t_{best} = \{\}$
- Let M be the number of attributes of the data structure DS
- Define the maximum runs to $TMAX$ and set the current number of runs to $nt = 0$
- Initialise the current start time $tstart$, set $t = tstart$ and the number of shipments to n

Step 1 (Setup/updating step):
- For each shipment $i = 1,...,n$ construct, or update if $t > tstart$, the set of promising shipments for consolidation $\mathbf{N_i}$ (see Section 5.2).
- Determine, or update if $t > tstart$, the 2-dimensional logical matrix
$$OLDFLAG = \text{OLDFLAG}_{ij}; i,j = 1,...,n$$
- Construct, or update if $t > tstart$, the data structure $DS = (DS_{im})_{i=1,...,n;m=1,...,M}$ (see Section 5.1)
- Initialise the 2-dimensional logical matrix $FLAG$ to
$$FLAG_{ij} = OLDFLAG_{ij}; i,j = 1,...,n$$
- initialise the temporary data structure DS as $TDS^t_{im} = DS_{im}; i = 1,..,n; m = 1,...,M$

Step 2 (Construction of the solution at time t):
- Generate uniformly two random numbers $\alpha \in [0.05, 0.5]$ and $\beta \in [0,1]$
- Construct RCL (see Subsection 4.3.2) as follows
  If $\beta \leq 0.5$ construct the RCL based on those unassigned pairs whose saving is within
  $(1+\alpha)$ percentage deviation from the current best saving
  Otherwise, construct the RCL based on those unassigned pairs whose regret is within
  $(1+\alpha)$ percentage deviation from the current maximum regret

- Select the pair of shipments randomly from RCL, say (i_*, j_*), set $TS = TS + S_{i_* j_*}$ and

$$E^t = E^t \cup \{(i_*, j_*)\}$$

- If there are still feasible pairs of shipments to consolidate
  Set $FLAG_{i_* j_*} = false$ and remove all combinations with shipments in E^t by setting
  $FLAG_{i_* j} = false$ for all j and $FLAG_{ij_*} = false$ for all i.
  If $i \notin E^t$ and i is affected, then update TDS^t_{ik} (see Section 5.1) and go back to Step 2.
  Otherwise, record the total saving TS and the consolidated pairs in E^t

Step 3 *(Local Search):*
    Apply the local search *LS* (see Subsection 4.3.3), record *TS* and the solution configuration E^t.

Step 4 *(Stopping criterion for the solution generated for period t)*
    If $nt \leq TMAX$

<div style="border:1px solid">

If $TS \geq TS_{best}$ set $TS_{best} = TS$ and record the best solution configuration $E_{best}^t = E^t$.

Set $nt = nt + 1, TS = 0$, $TDS_{im}^t = DS_{im}; i = 1, .., n; m = 1, ..., M$,

$FLAG_{ij} = OLDFLAG_{ij}; i, j = 1, ..., n$ and go to Step 2.

Otherwise

Display the best configuration and the overall best total saving TS_{best} for this period t.

Select from the chosen solution, those shipments that are to be physically assigned to

common carriers, say $\eta \leq n$, remove these from further investigation and go to Step 5.

Step 5 *(Roll Over):*

Set $t = t + \Delta$, $E^t = \{\}$, $nt = 0, TS = 0$, $TS_{best} = 0$ and $E_{best}^t = \{\}$.

Let n' be the new shipments arriving at time Δ, set $n = n - \eta + n'$ and go to Step 1

</div>

Figure 8: A Real Time GRASP-Based Meta-Heuristic

4.3.2 Construction of RCL (Step 2 of Figure 8)

As mentioned earlier, at each run, we incorporate two aspects which we believe to be interesting

(i)   the random choice of the objective function to be used within RCL and

(ii)   the variable size of RCL.

In (i), we select a random number $\beta \in [0,1]$ uniformly and we base our threshold depending on the objective function chosen, namely, the saving or the regret. This is performed as follows:

If $\beta \leq 0.5$ the saving criterion is used, otherwise the regret cost is considered instead.

In (ii), we select a random number α from a range $[\alpha_{min}, \alpha_{max}]$ with $0 < \alpha_{min} < \alpha_{max} < 1$. In our experiments setting $\alpha_{min} = 0.05$ and $\alpha_{max} = 0.5$ showed to be promising. This concept of varying the size of the RCL within each run provides extra flexibility to the search. This procedure continues until there is no more feasible shipments to consolidate.

4.3.3 Local Search (Step 3 in Figure 8)

Once a solution configuration is obtained, a local search which we refer to *LS* is activated. A solution configuration is defined as $\{(i_1, j_1), (i_2, j_2), ..., (i_p, j_p)\} \cup \{i_1', ..., i_q'\}$ where $2p + q = n$. The

first subset is made up of p pairs being consolidated while the second subset consists of singletons that are left unchanged (*i.e.*, $q = n - 2p$).

The move of the LS is defined by randomly choosing a pair (i_v, j_v) from a uniform distribution to be assessed against either (a) another pair (i_w, j_w) or (b) a singleton both of which are also chosen randomly and uniformly.

In (a), the swapping is performed by exploring whether the new pairing (i_v, j_w) and (i_w, j_v) yields a better saving. In other terms, if $Max(S_{i_v j_w} + S_{i_w j_v}, S_{i_v i_w} + S_{j_v j_w}) > S_{i_v j_v} + S_{i_w j_w}$, the pairs are swapped accordingly.

In (b), the pair (i_v, j_v) is exchanged against one of the singleton i_w'. If $Max(S_{i_v i_w'}, S_{i_w' j_v}) > S_{i_v j_v}$, one of the shipments i_v or j_v is swapped with i_w' to make up the new pair. The above operations are performed only when feasibility is maintained. The process is repeated using L feasible attempts and the best solution is selected. In our experiments, we set $L = \max(100, n)$. In other words, this strategy sits between the best and the first improvement as commonly used in the metaheuristic literature (see Salhi, 2017).

5 Speed Up Mechanisms

It is worth mentioning that as the problem is to be solved in real time and given the shippers' urgency to receive the quote, the entire process needs to be rather fast. Here, at each time period $t + \Delta$ (here Δ is set to 1 minute), both the running of the algorithm and the updating make up the total computational time. The update (see Steps 1, 4 and 5 of Figure 8) consists of removing any shipments that were agreed with the common carriers, the systematic inputting of new calls with their corresponding characteristics and the updating of both the data structure and the neighbourhood reductions.

To achieve this outcome, two important speed up schemes that contribute to the efficient implementation of the algorithm are presented. This includes (a) the design of a data structure and (b) the construction of neighbourhood reduction schemes. These two aspects are explored in

Subsections 5.1 and 5.2 respectively. Efficient data structures and neighbourhood reductions are initially developed for routing problems by Osman and Salhi (1996) and Salhi and Sari (1997). Similar speed up mechanisms are recently strengthened by Sze, Salhi and Wassan (2016, 2017) to solve efficiently large routing instances.

5.1 Data Structure Design

To avoid re-computing information that is already computed from one run to another, the following data structure that stores the useful attributes for each shipment $i, i = 1, ..., n$ is designed. The power of an effective data structure will cut down on computational burden drastically though it may require a small extra storage space. The idea is to build an intelligent system that records computed data that does not need to be recomputed again. It is important to identify these key points as these can be problem specific.

The data structure, DS, is defined by its elements DS_{im} which represents the mth attribute ($m = 1, ..., M$) for the ith shipment ($i = 1, ..., n$) with M being the number of attributes used. In this study, we have $M = 13$ attributes defined as follows;

DS_{i1} : the shipment that yields the best saving to consolidate with shipment i

DS_{i2} : the configuration that corresponds to the best saving to consolidate with shipment i

DS_{i3} : the total distance for the initial configuration (i.e., $\pi^0_{i,DS_{i1}}$)

DS_{i4} : the total distance for the best configuration (i.e., $C^1_{i,DS_{i1}}$)

DS_{i5} : the best saving for the best configuration (i.e., $S_{iDS_{i1}}$)

DS_{i6} : vehicle types used (1 to 9) for the best configuration

DS_{i7} : the shipment that yields the second best saving to consolidate with shipment i

DS_{i8} : the configuration that corresponds to the second best saving

DS_{i9} : total distance of the initial configuration with respect to the second best (i.e., $\pi^0_{iDS_{i7}}$)

DS_{i10} : total distance for the second best configuration (i.e., $C^1_{i,DS_{i7}}$)

DS_{i11} : the best saving for the second best configuration (i.e., $S_{iDS_{i7}}$)

DS_{i12} : vehicle types used (1 to 9) for the second best configuration

DS_{i13} : regret cost (i.e., $\underset{j}{Max}(RG_{ij})$)

The data structure DS is constructed and updated in Step 1 of Figure 8. As the approach is run in real time, we initialise a copy of DS in a temporary storage TDS which is updated whenever a pair of shipments is selected. Note that TDS_{im} ; $m=1,...,M$ is only updated when $i \notin E^t$ and $FLAG_{ij} = True$ for those j affected (i.e., $j = TDS_{i1}$ or $j = TDS_{i7}$.) Once the display of the consolidated shipments is given at period t, the necessary information is updated in Step 4 by creating again a copy of DS and $FLAG$ as TDS and $OLDFLAG$ respectively. Finally, the search continues for the next period by resetting the appropriate information in Step 5 and then reverting back to Step 1.

5.2 Neighbourhood Reduction Schemes for Order Consolidation (Case of Scenario 1)

It is equally important to distinguish between shipments that could be consolidated and the others that are unlikely to do so. Note that this guidance is introduced in addition to those shipments that violate feasibility (see Subsection 3.3). Given that not all feasible combinations are worth exploring, it is therefore important to identify the promising ones so to avoid performing unnecessary operations. The following neighbourhood reduction schemes are designed to respond to this challenge.

Definition- Two shipments are considered potential for consolidation for Scenario 1 (see Subsection 3.3 (a)) if at least one of the following two conditions is satisfied

        (1) Their collection points (delivery points) are within a certain threshold

        (2) One of the two shipments is en-route of the other (via collection, delivery or both)

Let $RMAX$ be the radius of proximity for which two collection (delivery) points can be served by the same vehicle. Similarly let $DMAX$ be the threshold denoting the additional extra route distance allowed for a vehicle to do a detour to pick up the other shipment on its way. Shipments i and j are considered to be candidates for consolidation, and hence $FLAG_{ij}$ is set to $TRUE$, if at least one of the following four rules is satisfied.

Rule 1: $d_{C_iC_j} \leq RMAX$ & $d_{D_iD_j} \leq RMAX$ (16a)

Rule 2: $d_{D_iD_j} \leq RMAX$ & $d_{C_iC_j} + d_{C_jD_i} - d_{C_iD_i} \leq DMAX$ (16b)

Rule 3: $d_{C_iC_j} \leq RMAX$ & $d_{C_iD_j} + d_{D_jD_i} - d_{C_iC_j} \leq DMAX$ (16c)

Rule 4: $d_{C_iC_j} + d_{C_jD_j} + d_{D_jD_i} - d_{C_iD_i} \leq DMAX$ (16d)

In Rule 1 (16a), both collection and delivery points are within a proximity distance set as $RMAX$ whereas in Rule 2 (16b), the delivery points are within proximity but the collection points are not though shipment j has its collection point C_j en-route from C_i to D_i. This insertion type rule could be modified to use the nearest of the two deliveries instead of D_i in the calculation.

Rule 3 (16c) is similar to (16b) except that the deliveries are not within proximity but D_j is en-route going to D_i. Rule 4 (16d) treats the case when neither collection nor delivery are in proximity but $C_j - D_j$ are en-route from C_i to D_i.

In summary, we define the reduced neighbourhood of each shipment $i\,(i = 1,...,n)$ as

$$\mathbf{N_i} = \{(C_j, D_j); j = 1,...,n \,|\, j \text{ satisfies (16a) } \vee \text{ (16b) } \vee \text{ (16c) } \vee \text{ (16d)}\}$$

In other words, for each shipment $i\,(i = 1,...,n)$, we only consider those shipments that belong to $\mathbf{N_i}$. This neighbourhood reduction scheme, which borrows ideas from expert knowledge, avoids evaluating non promising alternatives resulting in non performing unnecessary evaluations making the entire scheduling/optimisation tool much more effective and practically useful.

The primary goal of these rules is to restrict the consolidation for other shipments as highlighted in Scenario 1 of Subsection 3.3(a) only. The use of transshipment points for consolidation is unaffected even where the above four rules are not satisfied. It is also worth noting that though both speed up schemes cut on the computational effort considerably, however, if neighbourhood reductions are not designed appropriately resulting in removing some promising solutions, this could affect the quality of the solutions. This risk is not present when using a data structure as this is based on the efficient recording of useful information that do not need to be re-evaluated again. For a general discussion on these two important aspects, see chapter 6 in Salhi (2017).

6 Computational Results

The proposed approach is coded in C++ and run on an Intel (R) Core i7 2.4 GHz PC with 16GB RAM. We first provide a small illustrative real life example. To assess the performance of the proposed GRASP implementation, a comparison of the results against ILOG CPLEX is given. Real life based experiments including a case study are then tested using GRASP against the commonly used approach by the 3PL company.

6.1 Illustrative example

Figure 9a shows a real life example as part of the large network of the daily company operations where two separate shipments happen to be selected by the GRASP approach. These two shipments are requested with urgent collections and deliveries for two small packages from their respective shippers. Shipment 1 has a collection in Budapest, Hungary to be delivered to Cologne in Germany whereas shipment 2 needs also to collect from Budapest to deliver in Desteldonk, Belgium. Given the high volume of work for each operator at the 3PL company, these two requests are unlikely to be received by the same operator and as the two separate shippers require urgent quote from the operators, this will consequently lead to using two separate vehicles, each organised by one operator. The 3PL company strives in providing a high level of customer service to the shippers in resolving their problem very quickly though the delivery itself may take many hours.
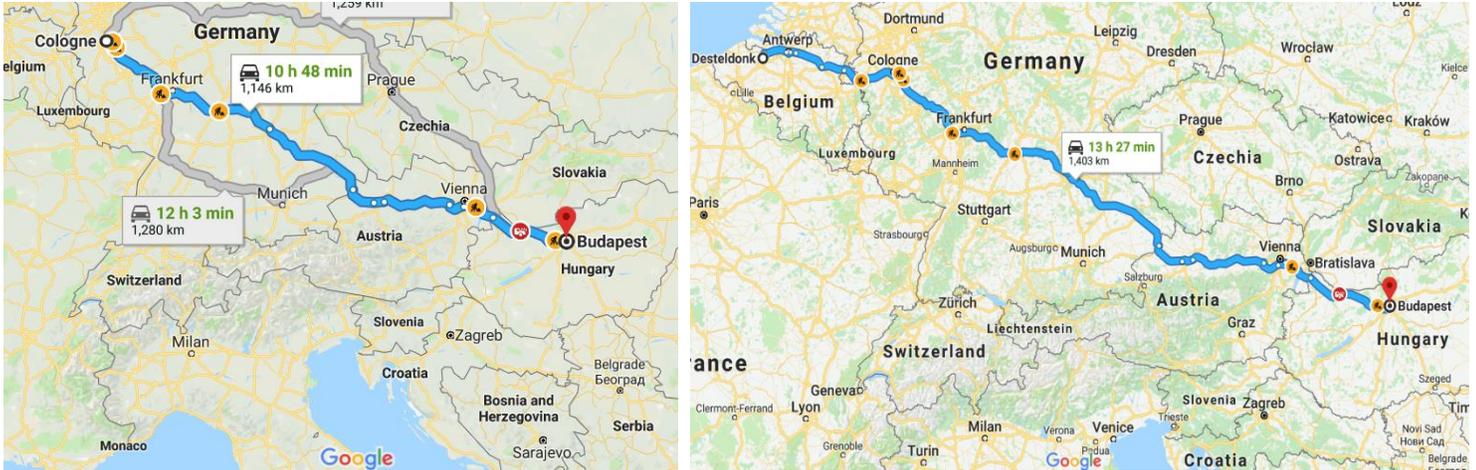
Figure 9a: Company's traditional Schedule of these two separate orders

The proposed decision support system deployed within the real-life setting captures over 100 live requests and presents potential options for the request using the speed of GRASP. Among the chosen pairs, for this particular example, a solution was ultimately made between both shipments. The original schedule is to send two separate vans for a total mileage of 2549 km and a total travel time of 24 hrs 15mins. The new schedule is achieved by using one van that combines both journeys starting from Budapest where both collections are taken, have a small detour to deliver at Cologne then continue to Desteldonk. This is displayed in Figure 9b. This chosen configuration corresponds to configuration [2] in Table 1. This new configuration consumes a total distance of 1418 km and travel time of 13hrs 44mins. In other words, for this particular example, a massive 1131 km of travel can be avoided leading to a cost saving of over €300 and a reduction of 0.29 tonne of CO_2 emission where the computation of the CO_2 emission is performed as follows:

Computation of CO2 Emissions

The amount of CO_2 emissions between locations i and j is defined as $\mu_1 \mu_2 d_{ij}$ with μ_1 and μ_2 denoting, for a given type of truck, the number of litres of Diesel consumed per km and the amount of CO_2 emissions in tonne per litre respectively. More details can be found in Green house report-UK (2017). As vans are used in this example (the smallest vehicle that can accommodate both shipments), the following parameter values are adopted $\mu_1 = 0.097$ and $\mu_2 = 0.0263$.
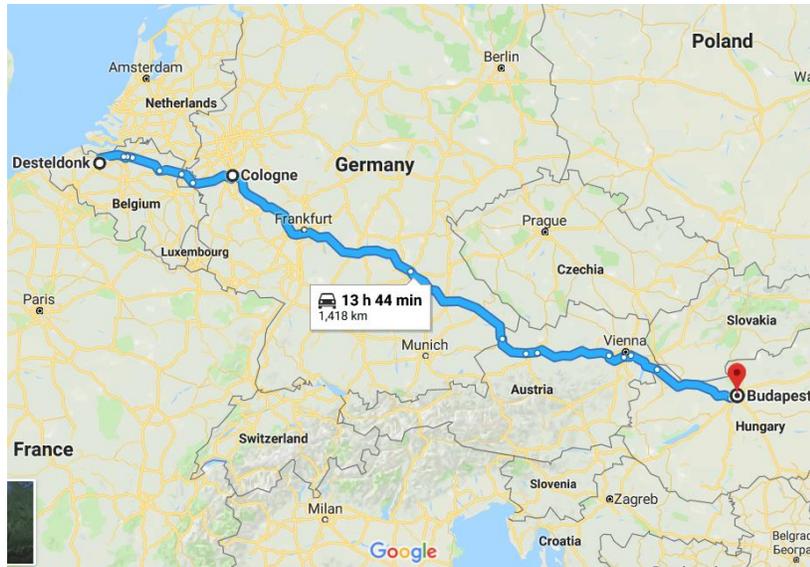
Figure 9b: An illustration of a consolidated solution by the GRASP approach

6.2 Performance Evaluation of GRASP vs CPLEX

To assess the performance of GRASP, we tested it against the exact method formulated in Subsection 4.1 using CPLEX 12.6 on four small instances with 20, 25, 30 and 60 shipments each.

These are collected as small samples from the company previous daily requests in the summer 2017, most of which originated from continental Europe destined to the UK. Though these instances are manageable for our formulation, due to our simple transformations, the larger ones (say, $n = 60$) would be too big to solve if a classical PDP type formulation is used instead.

The characteristics regarding individual shipments (collection/delivery, time windows, dimensions, etc.) and potential transshipment points are confidential but can be collected from the authors if need be. Table 2 summarises the distance travelled, the amount of CO_2 emission and the CPU time (in seconds) for both methods.

It can be observed that GRASP provides equal or very similar results as the optimal solutions, besides requiring a fraction of the time of CPLEX especially when the size of the problems gets larger as it is usually this real life case study. For example, GRASP is nearly 3 times faster for $n = 60$ which is relatively small in practice. Although CPLEX time does not seem to be excessive, this is still not practical in real time decision making as the instances are often much larger (at least

100 shipments) leading to unacceptable waiting time for the operators. Besides, the metaheuristic GRASP has the advantage of being flexible and easy to modify and to control if necessary. For instance, adding extra complexity to the problem or reducing the number of runs can easily be implemented if need be.

Table 2: Results using CPLEX and the GRASP approach

Instance #	# order requests (n)	GRASP			CPLEX		
		Distance Saving (km)	CO_2 Saving (tonne)	CPU (secs)	Distance Saving (km)	CO_2 Saving (tonne)	CPU (secs)
1	20	2,712	.69	.09	2,712	.69	.16
2	25	9,576	2.44	.10	9,860	2.52	.23
3	30	3,743	0.95	.19	3,743	0.95	.30
4	60	14,751	3.76	.81	14,858	3.79	2.3

It is also worth mentioning that the effect of the speed up mechanisms presented in Section 5 is considerable yielding on average around 70% reduction in computing time. It is found that the increase in computing time is approximately linear with the number of requests making the search very effective. For simplicity, the details are not reported here as this observation is in line with earlier studies such as the recent work by Sze *et al.* (2016), who show that the effect is statistically significant.

6.3 Performance Evaluation of GRASP in Real Scenario Setting

In this subsection, we analyse the proposed approach under two scenarios, namely, (a) an extensive experiment based on simulated real life instances and (b) on a real life case study.

6.3.1   Extensive experiment on one week data (GRASP vs Manual Method)

 In this experiment, we chose five days (one week) data in the Autumn of 2017. This is used as a platform to test the proposed method under the complexity of the problem the company faces on a daily basis. Though each day relates to one instance, the order of the instances does not represent necessarily Monday to Friday but for convenience these are ranked in ascending order of the

number of shipments. The proposed approach is then compared against the manual method adopted by the operators. We demonstrate the usefulness of our approach through the reduction of cost and environmental impact measured in terms of reduction in CO_2 emission.

The company provides the list of potential transshipment locations with opening and closing times, site's maximum lifting capacity and cost per movement. As there are too many collection (delivery) points, a simple aggregation scheme is employed. Europe is split into several geographical regions $R_r \in \Re$ each one with its base or reference location point $\bar{R}_r$ representing that region (say major city). For each triplet of regions $\{R_r, R_s, R_l\}$, the set of transshipment points Ω_{rs}^l is identified where each transshipment point is defined by its four attributes as described in Subsection 3.4. In Table 3, we summarise the five instances with the main countries involved, and their respective number of collections and deliveries in brackets. The number of shipments varies from 100 to 150, originating from all over Europe with some occurring in an interval of 1 to 2 minutes.

Table 3: Summary information on the shipments per country with (*a, b*) refereeing to *a* as #collections and *b* as # deliveries

Inst-ance #	# order requests	Belgium	Czech Republic	Germany	Spain	France	United Kingdom	Hungary	Italy
1	100	(7,14)	(21,6)	(18,28)	(27,0)	(14,7)	(0,38)	(13,0)	(0,7)
2	105	(7,4)	(17,8)	(23,38)	(43,0)	(4,12)	(0,38)	(11,0)	(0,5)
3	115	(11,9)	(13,17)	(33,24)	(12,23)	(16,13)	(23,19)	(5,8)	(2,2)
4	130	(11,9)	(19,21)	(31,31)	(31,11)	(6,14)	(20,28)	(11,12)	(1,4)
5	150	(11,10)	(22,9)	(31,56)	(63,3)	(7,12)	(0,55)	(15,1)	(1,4)

The results are presented in Table 4 following a similar format as in Table 2 except that CPLEX is replaced by the company's existing approach (company schedule or the manual one). The 4th column (*Co,Tr*) under GRASP is added to highlight the impact each of the two consolidation schemes may produce, with *Co* being the number of pairs found through order consolidation while *Tr* is the consolidation of shipments obtained through transshipment locations.

Table 4: Comparison of the results of the GRASP approach and the company's existing schedule

Instance #	# order requests (n)	GRASP				Company Schedule	
		Distance (km)	CO_2 (tonne)	CPU (secs)	(Co,Tr)	Distance (km)	CO_2 (tonne)
1	100	62916	16.06	1.43	(37,9)	109737	28.00
2	105	125047	15.85	1.42	(37,5)	221387	56.48
3	115	269592	17.28	1.26	(45,7)	471243	120.22
4	130	203341	19.97	1.30	(40,8)	356482	90.94
5	150	90490	23.08	1.56	(63,4)	171123	43.66
Total	600	**361552**	**92.24**		**(222,33)**	644488	164.42

The results over the five instances (five days) show around 78% reduction is obtained in both cost and CO_2 emission. These are given in bold. This is achieved by pairing 444 shipments out of 600 through 222 order consolidations and 66 via 33 transshipments. In brief, this accounts for a total of 510 out of the 600 shipments (i.e., a massive reduction of 85%) being consolidated leading to a significant increase in operation efficiency of the 3PL company while retaining its reputation as an outstanding service provider.

For illustration purposes, a chart showing the deviation (in %) from the company schedule for CO_2 emission is also given in Figure 10.
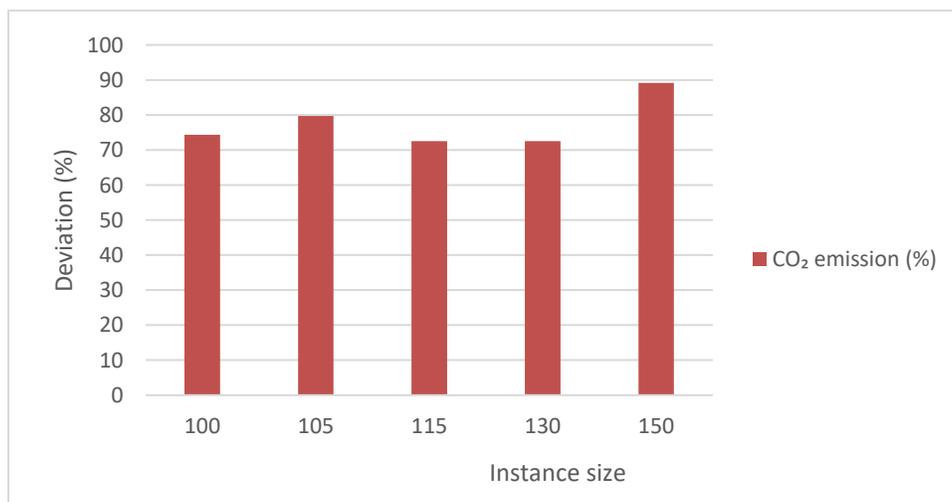
Figure 10:   Deviation (%) of CO_2 emission between the GRASP and the company results

As the cost saving and reduction in CO_2 emission are equivalent measures given the way CO_2 emission is calculated, we only represent one of the two, for example the latter in this case. The deviations are computed as follows:

$$\text{DevCost}(\%) = 100 * ([\text{cost(company)-cost(GRASP)}]/\text{cost(GRASP)})$$

$$\text{DevCO}_2(\%) = 100 * ([CO_2\text{(company)-CO}_2\text{(GRASP)}]/CO_2\text{(GRASP)})$$

It can be noted that a minimum of 72% reduction in CO_2 emission can be observed, with an average of over 77%. The same amount of cost saving is obviously recorded. This demonstrates the impact such a tool can have in assisting the operators in taking better decisions more efficiently and much faster resulting in gaining more quotes and hence increasing the competitive advantage of the 3PL company.

6.3.2 Case study

To assess the real performance of the decision support tool, starting in the week commencing the 4th of December 2017, the scheduling tool was applied with the company's logistics team where weekly cost savings were recorded for 13 weeks.

The cost savings was calculated by taking the savings of the tool-generated solutions identical to the team's implemented solutions. It was found that the average cost savings were around €2,000 a week. This means that if the team were not able to identify any solutions, due to an oversight or lack of experience, the tool would continue to highlight these solutions in order for the team to implement them. In Figure 11, we record an up-to-date track of the cost savings for this 13-week case study.

These results demonstrate that in some weeks such as weeks 6, 9 and 12 there was a relatively higher number of operations (busy periods) resulting in a relatively larger saving. This is because the operators, in these busy periods, find the tasks of identifying order consolidations (either via en-route or through transshipments points) very quickly relatively harder and overwhelming. This natural human related drawback does not feature within the expert system whose behaviour is retained throughout the periods.
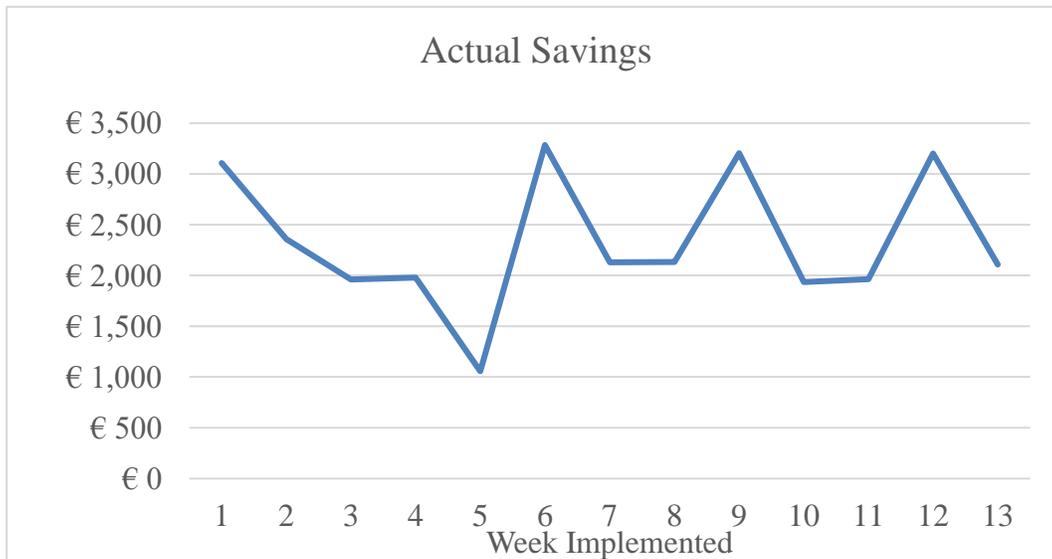
Figure 11: Actual cost saving for the 13 weeks period starting on the 4th of December 2017

Overall, there was positive feedback from the company management about the decision support tool, mainly due to its simplicity, speed and accuracy in providing sensible solutions. Additionally, as a by-product, the tool indirectly promoted new habits in embracing new technology and innovation.

7 Conclusion, Limitations and Suggestions

In this study, we highlight the emerging area of logistics, namely, time-critical freight logistics, also known as operational contingency logistic, where bidding for shipments need to be agreed between the shippers and the 3PL operators relatively quickly. This is in contrast to ordinary planned distribution where shipments are first gathered and an efficient computerised distribution system is applied and a bid produced. Here, efficiency can also be sought through innovative and intelligent tools that are adapted accordingly to fit this type of real time logistical problems using expert knowledge. Consolidation of pairs of shipments is explored using order consolidation and the introduction of newly located transshipment points whichever is most profitable. The best saving is then computed for every pair and an efficient matching is mathematically formulated and solved using an effective implementation of GRASP enhanced by two new attributes that provides extra flexibility in the construction of the restricted candidate list (RCL). To speed up the process given that the problem is in real time, data structures and neighbourhood reduction mechanisms

41

are developed with the help of expert knowledge resulting in an effective and intelligent expert system.

It is found that over 76% reduction in mileage and CO_2 emission can be obtained while maintaining high customer service level. These empirical results demonstrate that the proposed approach makes an interesting and effective addition in assisting the 3PL company when dealing with their real time scheduling issue. This optimization/scheduling tool could be made even more important when some of the offices of the company are merged resulting in a larger pool of shipments which could lead to an even higher level of efficiency in the company operations. This massive gain can be measured in terms of a reduction in operating cost leading to additional bids being accepted while retaining or even enhancing customer service, and consequently as a by-product contributing to a cleaner environment.

Limitations- our approach has some limitations that are worth mentioning.

(a) The system relies on importing live data from an external source which can be a handicap in some situations. One way forward tis to convince the company to integrate the optimisation tool into their entire IT system so to speed up the process even further.

(b) Though the response time to the customers needs to be done urgently, the implementation of the schedules are less strict. The process can be performed into two stages where in stage 1, an even faster tool that first approximates the gain instead could be developed resulting in a hopefully attractive quote. This is then followed in phase 2 by our GRASP-based tool which will then aim to achieve such a target or even better it.

(c) In this case study we convert the mileage into CO_2 emission through a linear relationship and some appropriate coefficients based on the truck sizes which are given in the Green house report (2017). This transformation be revisited by incorporating powerful analytical model that uses not only the size of the trucks but also the speed and the load of the trucks.

(d) The selection of the transshipment point for two shipments is based on (5a-5d). For instance, equations (5a) and (5b) could be modified to cater for the full insertion cost of the transshipment point but for simplicity and given the location of the collection and delivery points that the company deals with, in this occasion the sum of the distances from the collection points to the transshipment point is used only. A similar modification could also be adopted for (5c) and (5d).

(e) Another limitation, which is common to many logistic companies, is the initial willingness of the operators to embrace the change by integrating the new tool into their daily operations. This was easily and quickly resolved as we developed good relationships with top management as well as the operators which then made the entire project relatively much smoother to run.

Suggestions- Our study could also be extended in the following research directions.

(i) We could examine catering for more than a pair of shipments for consolidation while maintaining the current level of customer service. For instance, exploring the possibility for consolidation with three or four shipments with not necessarily one transshipment only could be worthwhile. The mathematical formulae that enumerate the configurations will need to be derived accordingly. The number of possibilities will increase drastically from 8 as used here to a much larger number. We think these pre-computations of the cost saving, though slighter larger in terms of computation, will still outweigh by far the practical implementation of the general ILP formulations.

(ii) In this study, we opted for a trajectory (or single solution) type metaheuristic like GRASP. Other well established and powerful trajectory methods such as Variable Neighbourhood Search (VNS) and Large Neighbourhood Search (LNS) could also be adopted or even hybridise to produce even better solutions. Evolutionary methods, also known as population-based approaches, such as Genetic Algorithms, Particle Swarm Optimisation or Ant Colony Optimisation, can also be considered and integrated with trajectory ones. A hybridisation of metaheuristics and exact methods known as matheuristics could also be investigated. As the latter two approaches may be too time consuming, the need for speedup rules to make the search fast enough to cater for real time issues is paramount.

(iii) The availability of the suppliers is taken for granted here as we assume there is always a common courier ready to collect and deliver the shipments. However, it would be interesting to integrate the supplier selection, as explored by Kaya and Yet (2019) for instance, into the scheduling/optimisation part.

(iv) Currently we base our search on existing orders even though these are continuously updated due to real time requirement. One way forward would be to generate, in addition to the current orders, probable orders using deep learning, statistics and expert knowledge

to forecast the location of the orders that are likely to arrive. This extra information could then be used in our GRASP metaheuristic, or a similar expert-based system, to build the schedules. This will obviously provide the search with a wider feasible region and hence better solutions could be obtained. However, one needs to be cautious as a penalty associated with orders that may not be realised needs to be incorporated into the model.

It is important to stress that though the above extensions are exciting academically and useful from a practical view point, one ought not to underestimate the considerable level of added complexity that exists. We are currently examining some of the above issues which we are confident will trigger other exciting research avenues.

Acknowledgements

References

1. Abrache, J., Crainic, T.G., Gendreau, M., Rekik, M. , 2007. Combinatorial auctions. *Annals of Operations Research* 153, 131-164.

2. Agatz, N., Erera, A., Savelsbergh, M., Wang, X., 2011. Dynamic ride-sharing: a simulation study in metro Atlanta. *Transportation Research Part B* 45, 1450-1464.

3. Al-Ghami, Z., Manier, H., Manier, M.A., Chebib, E., 2018. An Advanced GRASP-HGA combination to solve a multi-period Pick and Delivery problem. *Expert Systems with Applications* 205, 262-272

4. Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G., 2007. Static Pickup and Delivery Problems: A Classification Scheme and Survey. *TOP* 15, 1-31.

5. Campbell, J.F., 1990. Freight consolidation and routing with transportation economies of scale. *Transportation Research Part B* 24, 345-361.

6. Chang, T.S., 2009. Decision support for truckload carriers in one-shot combinatorial auctions. *Transportation Research Part B* 43, 914-933.

7. Cortes, C.E., Matamal, M., Contardo, C., 2010. The pickup and delivery problem with trasnfers: formulation and a branch-and-cut solution method. *European Journal of Operational Research* 200, 711-724.

8. Danloup, N., Allaoui, H., Goncalves, G., 2018. A comparison of two meta-heuristics for the pickup and delivery problem with transshipment. *Computers & Operations Research* 100, 155-171.

9. Estrada-Romeu, M., Robuste, F., 2015. Stopover and hub-and spoke shipment strategies in less-than-truckload carriers. *Transportation Research Part E* 76, 108-121.

10. European Environment Agency., 2018a. GHG emissions by sector in the EE-28 (1990-2016), updated in march 2018.   https://www.eea.europa.eu/data-and-maps/daviz/ghg-emissions-by-sector-in#tab-chart_1 (accessed on 18/12/2018)

11. European Environment Agency., 2018b. Energy consumption by mode of transport in EEA33 (1990-2016), updated in August 2018, https://www.eea.europa.eu/data-and-maps/indicators/transport-final-energy-consumption-by-mode/assessment-9 (accessed on 19/12/2018)

12. Feo, T.A., Resende, M.G.C., 1989. A probabilistic heuristic for a computationally difficult set covering problem, *Operations Research Letters* 8, 67-71.

13. Feo, T.A., Resende, M.G.C., 1995. Greedy Randomised adaptive search procedure, *Journal of Global Optimization* 6, 109-133.

14. Furuhata, M., Dessouky, M., Ordez, F., Brunet, M.E., Wang, X., Koeing, S., 2013. Ridesharing: the state-of-the-art and future directions. *Transportation Research Part B* 57, 28-46.

15. Gendreau, M., Potvin, J.Y. 2019. Handbook of Metaheuristics (updated edition). New York: Springer.

16. Green house report-UK.   2017. https://www.gov.uk/government/publications/greenhouse-gas-reporting-conversion-factors-2017. (accessed on 18/12/2018)

17. Hall, R.W., 1987. Consolidation strategy: inventory, vehicles and terminals. *Journal of Business Logistics* 8, 57-73.

18. Kaya, R., Yet, B., 2019. Building Bayesian networks based on DEMATEL for multiple criteria decision problems: A supplier selection case study. *Expert Systems With Applications* 134, 234-248.

19. Lenstra, J. K., Kan, A. H. G. R. , 1981. Complexity of vehicle routing and scheduling problems, *Networks* 11, 221- 227.

20. Lu, E.F.C., Yang, Y.W., 2019. A hybroid route planning approach for logistics with pickup and delivery. *Expert Systems With Applications* 118, 482-492.

21. Masson, R., Lehuede, F., Peton O., 2013. An adaptive large neighbourhood search for the pickup and delivery problem with transfers. *Transportation Science* 47, 344-355.

22. Masson, R., Ropke, S., Lehuede, F., Peton O., 2014. A branch-and-cut-price approach for the pickup and delivery problem with shuttle routes. *European Journal of Operational Research* 236, 849-862.

23. Mesa-Aranga, R., Ukkusuri, S.V., 2013. Benefits of in-vehicle consolidation in less than truckload freight transportation operations. *Transportation Research Part E* 60, 113-125.

24. Mitrovic-Minic, S., Laporte, G., 2006. The pickjup and delivery problem with time windows and transshipment. *INFOR: Information Systems and Operational Research* 44, 217-227.

25. Osman, I., Salhi, S., 1996. Local search strategies for the vehicle ,fleet mix problem, In V. Rayward-Smith, I. Osman, C. Reeves and G. Smith (Eds), *Modern heuristic search methods* (pp. 131-154). Wiley.

26. Parragh, S. N., Doerner, K. F., Hartl, R. F., 2008. A survey on pickup and delivery problems Part I: Transportation between customers and depot, *Journal für Betriebswirtschaft* 58, 21-51.

27. Qu, Y., Bard, J.F., 2012. A GRASP with adaptive large neighbourhood search for pickup and delivery problems with transshipments. *Computers and Operations Research* 39, 2439-2456.

28. Rais, A., Alvelos, F., Carvalho, M.S., 2014. New mixed integer programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research* 235, 530-539.

29. Resende, M.G.C., Ribiero, C.G., 2019. Greedy randomized adaptive search procedures: Advances and Extensions. In M. Gendreau and JY. Potvin (Eds). *Handbook of Metaheuristics* (chapter 6; pp 169-220), New York, Springer.

30. Resende, M.G.C., Ribiero, C.G., 2016. Optimization by GRASP. New York: Springer-Verlag.

31. Salhi, S., 2017. *Heuristic Search: the emerging science of problem solving*. Switzerland: Springer.

32. Salhi, S., Sari, M., 1997. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem, *European Journal of Operational Research* 103, 95-112.

33. Shang, J.S., Cuff, C.K., 1996. Multicriteria pickup and delivery problem with transfer opportunity. *Computers and Industrial Engineering* 30, 631-645.

34. Sze, J.F., Salhi. S., Wassan, N., 2016. A Hybridisation of adaptive variable neighbourhood search and large neighbourhood search: application to the vehicle routing problem. *Expert System and applications* 65, 383-397.

35. Sze, J.F., Salhi, S., Wassan, N., 2017. The cumulative capacitated vehicle routing problem with min-sum and min-max objectives: An effective hybridisation of adaptive variable neighbourhood search and large neighbourhood search. *Transportation Research Part B* 101, 162-184.

36. Taha, H.A., 2017. Operations Research: An Introduction, Global Edition (10th edition), Pearson, NY.

37. Thangiah, S.R., Fergany, A., Awan, S., 2007. Real-time split-delivery pickup and delivery time window problems with transfers. *Central European Journal of Operations Research* 15, 329-349.

38. Ulku, M.A., 2012. Dare to care: Shipment consolidation reduces not only costs, but also environmental damage. *International Journal of Production Economics* 139, 438- 446.