DE GRUYTER OPEN

Information Technology and Management Science doi: 10.1515/itms-2015-0013

2015 / 18

Pickup and Delivery Selection: Problem Formulation and Extension to Problem Variants

Katrien Ramaekers[1], An Caris[2], Tabitha Maes[3], Gerrit K. Janssens[4]
[1, 2, 3, 4] *Hasselt University*

***Abstract* – Operational planning decisions of a carrier consist of accepting transport requests and constructing vehicle routes. However, a carrier has only a limited capacity within his own vehicle fleet and can only serve a selection of clients. Transport requests are accepted only if they contribute to a higher total profit. This practical aspect is modelled with the Pickup and Delivery with selection of customers. A mixed-integer programming formulation is given. It is indicated how the problem is solved by means of a meta-heuristic, more specifically a tabu-embedded simulated annealing algorithm. A number of alternative settings may appear in practice. They are discussed in this article.**

***Keywords* – Heuristic algorithms, operations research, optimisation, vehicle routing.**

I. Introduction

The operational planning decisions of a carrier consist of accepting transport requests of customers and constructing daily vehicle routes. Several customers may have less-than-truckload requests to be transported between two specified locations: these customers are consolidated into vehicle tours by the carrier. However, a carrier has only a limited capacity within his own vehicle fleet. Therefore, the carrier can only serve a selection of customers. Transport requests of customers are accepted only if they contribute to a higher total profit. But many companies outsource their transport activities to reduce overhead costs. Logistics service providers are hired to execute the transportation services. Such companies are called freight carriers, which operate in the "hire and reward" mode. Operational planning problems relate to decisions, such as crew scheduling (assignment of crews to vehicles or transhipment facilities); empty balancing (preparation of the operations for the next planning period), and vehicle routing and scheduling (scheduling of the services for the pickup and delivery phases). A carrier faces the daily problem of optimal scheduling of the transport orders. Each day a carrier receives transport requests from customers, which have to be executed within a certain time period. To obtain a maximum profit, the carrier has to group certain orders and create an optimal sequence of paired pickup and delivery tasks. In literature, this problem is called a pickup and delivery problem (PDP). When a carrier has to decide whether to accept a certain request, the problem is defined as a Pickup and Delivery Selection Problem (PDSP).

"Tactical request" acceptance problems require a general decision about the future acceptance of different requests. Mostly, this type comprises all requests of a certain customer. "Operational request" acceptance problems require that the carrier company has to decide about the acceptance of particular requests, which are not part of long-term contracts. Such a request is accepted if expected revenues cover expected additional costs caused by this additional request. If a carrier refuses a customer demand, it may be expected that also all other requests of this customer are lost for this carrier. The traditional PDP is extended to a PDSP by allowing a selection of transportation requests. This leaves the carrier with the option to discard transportation requests, which lead to a lower total profit. After planning and scheduling vehicles into routes as in a classical PDP, a selection within the transportation requests has to be made. The PDSP is formulated as a mixed integer linear program. The objective of the PDSP is to maximise the profit collected along the vehicle tours. Profit is defined as the sum of the total revenue collected on all the tours minus the total cost of performing the tours. A number of constraints are added. They relate to flow conservation constraints, vehicle constraints, time window constraints, and pairing and precedence constraints.

A number of alternative settings may appear in practice. Four of those settings are discussed: (1) Compulsory requests: due to long-term contract or for other commercial reasons, some of the requests cannot be neglected, even if they do not contribute fully to profit; (2) Fixed vehicle cost: for each vehicle that is executing a route a fixed vehicle cost is added to the total operating cost of the carrier. The carrier has to consider whether or not a vehicle will be performing a route; (3) Logistics service providers: in order to increase the number of transport requests that are accepted, a carrier may outsource requests to a logistics service provider (LSP). The consideration has to be made whether a request is performed with the vehicle fleet of the carrier, outsourced to an LSP or refused; (4) Alternative revenue model: the assumption is made that the price is fixed and that revenue depends on the distance between the pickup and delivery point of the customer. In most real life cases, transport prices depend on the total number of units transported and not only on the distance travelled.

II. Literature Review

The PDP is a generalisation of the vehicle routing problem (VRP) [1], [2]. In a VRP, generally all trip requests either originate or terminate at a single depot. In a PDP, the trip requests are made between two locations that are outside the depot. In this section, the division between paired and unpaired pickup and delivery points is used as in [3]. Pickup and delivery vehicle routing problems are characterised with

©2015 Katrien Ramaekers, An Caris, Tabitha Maes and Gerrit K. Janssens. This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License (http://creativecommons.org/licenses/by/4.0), in the manner agreed with De Gruyter Open.

unpaired pickup and delivery locations. In this case, an identical load is considered, and each unit picked up may be used to serve a delivery request. A classical pickup and delivery problem, on the other hand, has paired pickup and delivery locations. Every request is associated with a paired origin and destination location and a specified load.

The literature related to this specific case of the PDP, the pickup and delivery selection problem, relies on two main bodies of routing literature: on the one hand, the VRP with profits and, on the other hand, literature concerning PDP. A review of the literature on the VRP with profits and on the pickup and delivery selection problem is provided in [4].

This research offers the following novelties compared to existing research. The traditional PDP is extended to a PDSP by allowing a selection of transportation requests. This leaves the carrier with the option to discard transportation requests, which lead to a lower total profit. The problem at hand considers more than one commodity and paired pickup and delivery locations. This is different from the study of [5] where a single commodity is considered and pickup and delivery are unpaired. Furthermore, multiple vehicles are considered and transport loads are less-than-truckloads. In the study of [6], only a single vehicle is assumed and in [7]–[10] full truckloads are investigated.

The paired pickup and delivery locations together with the multiple vehicles and less-than-truckload requests make the PDSP very hard to solve. The only paper that studies a PDSP with similar problem characteristics but in a different problem context is [11]. Their heuristic results are not compared to exact solutions or lower bounds, and the reported results are briefly described. This hinders the comparison of computational results.

III. Problem Formulation

In this section, a mathematical representation of the problem is given. First, the key characteristics of a PDSP are described. Next, all symbols are introduced. Finally, the objective function and problem constraints are formulated. The problem is defined as a static PDSP problem. The formulation is an adaptation of the PDPTW formulation of [1].

A. Key Characteristics of PDSP

Not all requests have to be accepted, but every fulfilled request leads to revenue. If a request is accepted, a reward is achieved when the transport is done successfully. For every request, a hard time window is assigned to both the pickup and delivery location. A request is of the type "less-than-truckload". Furthermore, pickup has to occur before delivery of each request (Precedence constraint), and pickup and delivery have to be performed by the same vehicle (Pairing constraint). In the model, multiple vehicles are used of equal capacity. All vehicles depart from and return to a depot of the carrier. Finally, travel costs and travel times for each link are known and assumed to be constant.

B. Introduction of Symbols

Index Sets

$i \in P = \{1, \dots, n\}$: set of requests originating at the pickup locations.

$i \in N = P \cup D \cup O$: set of nodes where $P = \{1, \dots, n\}$ is the set of pickup locations, $D = \{n+1, \dots, 2n\}$ is the set of delivery locations, and O represents node 0, i.e. a single depot.

$k \in K = \{1, \dots, |K|\}$: set of identical vehicles.

Parameters

Q: uniform vehicle capacity.

ct: unit distance travel cost which is the same for each vehicle.

q_i: the quantity to be picked up or delivered at node $i \in N \backslash O$. It is positive if $i \in P$ and negative if $i \in D$.

Rev_i: the revenue earned from request $i \in P$.

d_{ij}: the distance between two nodes $i \; and \; j \in N$.

t_{ij}: the travel time between two nodes $i \; and \; j \in N$.

st_i: service time at node $i \in N \backslash O$. By convention the service time st_0 at the depot is zero.

e_i: the earliest service start time at node $i \in N \backslash O$.

l_i: the latest vehicle arrival time at node $i \in N \backslash O$.

s_k: the operation start time of vehicle $k \in K$.

f_k: the operation finish time of vehicle $k \in K$.

Decision Variables

Two sets of binary variables are defined: flow variables $\boldsymbol{X}$ and request acceptance variables $\boldsymbol{Y}$.

$X_{ij}^k = 1$ if vehicle k travels from i to j; 0 otherwise $(i, j \in N, k \in K)$.

$Y_i^k = 1$ if vehicle k performs request i; 0 otherwise $(i \in P, k \in K)$.

In addition to these binary variables, two sets of continuous variables are introduced to keep track of service completion times and vehicle loads.

T_i^k is service completion time at node $i \in N$.

L_i^k is load of vehicle k after serving node $i \in N$.

Requests. A carrier receives a set P of requests. Each request consists of a pickup location, a delivery location, a quantity to be shipped and a revenue if the request is completely satisfied. The quantity q_i may either be a positive or a negative number, depending on the type of operation, either a pickup or a delivery task.

Locations. Three different types of locations may be distinguished, each with their own time window. A set of pickup locations and a set of delivery locations are included, each with earliest operation time e_i, latest operation time l_i. A single depot O is available, where each vehicle starts and ends its route.

Network. Network $G(A,N)$ is given with $N = P \cup D \cup O$ – the set of nodes and A – a set of undirected arcs. Within the network, the distance between two nodes i and j is given as d_{ij}. The travel cost ct expresses the charge for travelling a single distance unit. The cost to travel a link is expressed as $ct.d_{ij}$. The variable t_{ij} represents the time needed to travel from node i to node j.

Vehicles. The carrier has a given homogenous fleet of own vehicles. Each vehicle k has capacity Q. Vehicles are bound in time by their driver due to legal driving time restrictions. Each vehicle has start time s_k and finish time f_k. To keep track of the

content of the vehicle, so that it does not exceed the capacity, load variables L_i^k are introduced.

Operations. A vehicle performs several operations on its route. Each pickup and delivery operation takes a certain amount of operation time ot_i to perform per unit that needs to be handled. The total time a vehicle spends at the pickup or delivery location is $ot_i \,.q_i$. A vehicle is allowed to arrive early at the location, but must then wait until the start of the time window. A vehicle is not allowed to finish its service at a location after time l_i. A drive-first strategy (drive first and wait at the arrival location) is used for the PDSP.

Objective Function

The objective of the PDSP is to maximise the profit collected along the vehicle tours. Profit is defined as the sum of the total revenue collected minus the total cost of performing the routes. The total revenue is found by accumulating all revenues of the requests that are accepted and executed.

$$Rev_{tot} = \sum_{k\in K}\sum_{i\in P} Rev_i . Y_i^k$$

The total cost (C_{tot}) is calculated as the sum of the costs of each link travelled by a certain vehicle k.

$$C_{tot} = \sum_{k\in K}\sum_{i\in N}\sum_{j\in N} ct \,. d_{ij} \,. X_{ij}^k$$

The objective function to be maximised is:

$$Profit = [Rev_{tot} - C_{tot}].$$

The formulation is different from a well-known formulation by [12] in the following way: (1) the objective function is different; (2) the non-linear constraints are linearised; and (3) variable T_i^k refers to the end of service rather than the start of service.

Constraints

Flow conservation constraints. These constraints are introduced to make sure that vehicles entering a location also leave the location.

$$\sum_{\substack{j=1\\ j\neq i}}^{N} X_{ij}^k - \sum_{\substack{j=1\\ j\neq i}}^{N} X_{ji}^k = 0\,, \forall i \in N, \forall k \in K$$

Vehicle constraints. Each vehicle starts and ends its route at the depot. An unused vehicle stays at the depot.

$$\sum_{j\in P} X_{0j}^k \leq 1\,, \forall k \in K$$

$$\sum_{i\in D} X_{i(2n+1)}^k \leq 1\,, \forall k \in K$$

Every request may be executed by at most one vehicle.

$$\sum\nolimits_{k=1}^{K} Y_i^k \leq 1, \forall i \in P$$

A vehicle cannot load more freight than its capacity.

$$L_i^k \leq Q_{max}, \forall k \in K, \forall i \in N$$

To keep track of the load of a vehicle at a certain moment, the following constraints are necessary. Each vehicle leaves from and returns to the depot empty.

$$L_0^k = 0, \forall k \in K$$

$$L_j^k - L_i^k - \left|q_j\right| \geq M_1.\left(1 - X_{ij}^k\right),$$
$$\forall i, j \in N \; and \; i \neq j, \forall k \in K$$

Time window constraints

Each node has to be served within its time window. The start of the operation, as well as the end of the operation has to fall within the time window.

$$e_i + ot_i.|q_i| \leq T_i^k \leq l_i, \forall i, j \in N \backslash 0, \forall k \in K$$

To keep track of time, a time variable is introduced. Initially, the time variable is set equal to the start time of the vehicle.

$$T_0^k = s_k, \forall k \in K$$

A vehicle may not exceed its finish time.

$$s_k \leq T_i^k \leq f_k, \forall i \in N, \forall k \in K$$

The arrival time at a node may not precede the earliest operation time allowed on that location. This is specified in the following constraint:

$$T_i^k + t_{ij} - T_j^k + ot_j \,.\left|q_j\right| \leq \left(1 - X_{ij}^k\right).M_2,$$
$$\forall i, j \in N, \forall k \in K$$

Due to the time window constraint on T_i^k, it is assured that the operation does not start before e_j.

Pairing and precedence constraints

If a request is performed, then vehicle k has to finish its operations at the pickup location i before it can visit the associated delivery location $n+i$. This is known as the precedence constraint expressed as:

$$T_i^k + t_{i(n+i)} - T_{n+i}^k \leq \left(1 - Y_i^k\right).M_2, \forall i \in P, \forall k \in K$$

It is not allowed to split a request over multiple vehicles. A vehicle has to perform both the pickup and the delivery operation. This is known as the pairing constraint expressed as:

$$\sum_{j\in N\backslash O} X_{ij}^k = Y_i^k\,, \forall i \in P, \forall k \in K$$

and

$$\textstyle\sum_{j\in N\backslash O} X_{j(n+i)}^k = Y_i^k\,, \forall i \in P, \forall k \in K.$$

In the expression, two big M-values are used, where M stands for a sufficiently large number. The value of M_1 is set equal to the capacity of the vehicles, and the value of M_2 is set equal to the maximum length of a working day.

IV. Solution Method

The problem under study is solved by means of a metaheuristic. The heuristic is based on the tabu-embedded simulated annealing algorithm of [13]. The algorithm starts

with an insertion heuristic to create the first feasible solution (S_{best}). This solution is further improved by an improvement heuristic. Instead of repeating the tabu search until the procedure terminates, it is restarted from the current best solution (S_{best}) after several iterations (*STOP*) without improvement. At the same time, the global annealing temperature T is reset. After a number of restarts K without improvement the algorithm is terminated. The generation of new best solutions (S'_{best}) is done via a tabu-search algorithm. The TSA algorithm is described below.

T(abu-embedded)S(imulated)A(nnealing) algorithm
Empty tabu set
Set *gNoImpr* = 0
Set *Sbest* to initial solution from insertion heuristic
Sbest = PostOptimize(*Sbest*)
Sbest = LocalSearch(*Sbest*)
while *gNoImpr* < *K* **do**
 S = *Sbest*
 S'best = TABU(*S*)
 if *Profit*(*S'best*) > *Profit*(*Sbest*) **then**
  Set *Sbest* to *S'best* and *gNoImpr* = 0
 else
  gNoImpr = *gNoImpr* + 1
  reset *T* to T_0
 end if
end while
output *Sbest*

The TSA algorithm makes use of some procedures called *PostOptimize*, *LocalSearch* and *TABU*.

The *PostOptimize* procedure relates to a REORDER operator. Starting from a solution, the operator tries to lower costs by reordering the nodes within existing routes. As no new requests are added, the revenue remains the same. Hence to increase profit, the cost of performing a route should decrease, i.e. when the total distance, the only cost driving factor, decreases. The operator tries to improve the original route by serving pickup points earlier in the route and delivery points later.

The *LocalSearch* procedure relates to an improvement heuristic in which four local operators are used to improve the initial solutions. Two operators are classical PDP operators. Two operators, called SHIFT and EXCHANGE, are similar to the local search operators of [13]. Both other operators are specifically designed for this heuristic. The INSERT operator tries to insert, from a list of non-served requests, a request in a route in a feasible way. If the insertion of such a request leads to a higher profit for the route considered, the request is inserted in the position with the lowest cost. The SWITCH operator removes the request with the lowest profit from the selected route and replaces it with a non-served request. These requests are switched if they result in a higher total profit.

The *TABU(S)* procedure aims to find from existing solution *S* a new local optimum (*S'local*) with the help of the SHUFFLE (*S*) algorithm. Random feasible solution (*S'*) is generated with the SHUFFLE(*S*) algorithm, which is not in the tabu list. This random solution (*S'*) is further optimised by means of the LocalSearch function. This leads to a new local optimum (*S'local*), which is compared to the current local optimum (*Slocal*). If it performs better, the new local optimum is stored in *Slocal* and transferred as an output of the TABU(*S*) algorithm. If the current local optimum (*Slocal*) is not improved, the search continues with the new found local optimum (*S'local*). The algorithm repeats itself until no improvement is found for a certain number of iterations (*STOP*). The pseudo-code of the Tabu Search procedure is given below.

Tabu Search procedure TABU(S)
Set S_{local} = *S*
Set *NoImpr* = 0
while *NoImpr* < *STOP* **do**
 S' = *SHUFFLE*(*S*), with *S'* not in tabu set
 S'_{local} = *LocalSearch*(*S'*)
 S'_{local} = *PostOptimize*(S'_{local})
 if *Profit*(S'_{local}) > *Profit*(S_{local}) **then**
  Set *Slocal* to S'_{local} and *NoImpr* = 0
 else
  NoImpr = *NoImpr* + 1
 end if
 Set *S* = S'_{local}
end while
output S_{local}

V. Variants of the Problem

A. Compulsory Requests

In some cases, it is possible to decide for each transport request whether it is fulfilled by the carrier or by a third party LSP. For high quality goods or because of reliability or trustiness, sometimes customers prohibit subcontracting. The customer requires fulfilment of the transport request by the carrier. Such a request is called a "compulsory request". Only limited research has concentrated on these kinds of requests, but important publications are [14] and [15].

The optimisation model has to be slightly adapted. Let index i^{comp} be used for the requests which are compulsory and i^{Ncomp} for those which are non-compulsory. The constraints to express this are as follows:

$$\sum_{k=1}^{K} Y_{i^{Ncomp}}^{k} \leq 1, \forall i \in P$$

$$\sum_{k=1}^{K} Y_{i^{comp}}^{k} = 1, \forall i \in P$$

To cope with compulsory requests, modifications to the TSA algorithm have to be made. The insertion heuristic is adapted as it is no longer obliged for each request to have a positive impact on the profit of a route. Furthermore, the insertion of compulsory requests has to be ensured. A list is created, which contains all compulsory requests. The insertion heuristic starts by placing the requests from this list into vehicle routes. Only the feasibility concerning time windows and vehicle capacity are taken into account. For the first compulsory request to be inserted into a route it is not checked whether the request is profitable. Subsequent compulsory

requests are inserted into the most profitable place in an existing route. This procedure leads to a higher overall profit or a lower loss within a route. Again, as in the original insertion heuristic, pickup and delivery are inserted separately at their best location keeping in mind the time precedence constraint. If a compulsory request cannot be inserted in an existing route at a profitable place, a new vehicle is added as long as vehicles are available. The first compulsory request of a new route may either lead to a profit or a loss. Further compulsory requests are inserted at the most profitable place in an existing route. When all available vehicles are in use and some compulsory requests remain non-served, they are inserted at the first available place regardless of the profit or loss. When all compulsory requests are inserted, the remaining requests are selected until all vehicles are fully used or no request can be inserted at a profit. Requests which are not compulsory are only inserted if the total profit increases.

B. Fixed Vehicle Cost

The cost evaluation of the routing plan consists of a variable part related to the distance or time travelled. This variable part might be different depending on whether each transport request is fulfilled by a carrier or by a third-party logistics service provider. But in addition to the variable part, also a fixed part may be included in the cost. The fixed vehicle cost represents the daily operating cost, the maintenance cost, and the labour cost. In case, a vehicle routing model does not assume that the fleet of vehicles is of a fixed size, it makes sense to associate a fixed cost on the use of a vehicle.

Decisions on the number of vehicles in the fleet of the carrier are tactical decisions and the cost of the vehicles is considered in the long run. Furthermore, it is supposed that a carrier has to pay its drivers regardless whether they operate a vehicle or not. This means that its cost structure does not change when a vehicle is used and, hence, no fixed vehicle cost is induced. In this section, this assumption is relaxed. For each vehicle that is executing a route, a fixed vehicle cost is added to the total operating cost of the carrier. The carrier has to consider whether or not a vehicle is performing a route. Only when the profit obtained from a route is high enough to cover the fixed vehicle cost, the route will be considered in the solution. This might lead to accepting fewer transport requests from customers.

In the mathematical programming model, the objective function needs to be modified. The total cost (C_{tot}) is calculated as the sum of the costs of each link travelled by certain vehicle k plus the fixed cost of each vehicle which is used (identified by the variable leaving the depot).

$$C_{tot} = \sum_{k\in K}\sum_{i\in N}\sum_{j\in N} ct\,.\,d_{ij}\;.X_{ij}^{k} + \sum_{k\in K}\sum_{j\in N} C_{veh}.X_{oj}^{k}$$

Also in this case the TSA algorithm is easily modified. First, the insertion heuristic and post-optimisation operators are run. Afterwards, the fixed cost of the vehicle is added to the cost of each of the constructed routes. In the third step, local search is applied to obtain the most profitable routes given the fixed vehicle cost. The fourth step checks whether all routes are profitable. A non-profitable route is removed and the requests from these routes are stored in a list. If one of the initially constructed routes is not profitable, fewer vehicles are used in the optimal solution. Next, the TSA algorithm is rerun taken the previous solution (with the unprofitable routes) removed as input. However, this time only the remaining vehicles are considered. For this reason, the requests removed from a deleted route together with the other non-served requests are considered for insertion in one of the remaining routes during the second run of the algorithm. As fewer routes are considered, the run time of the rerun is lower compared to the first run. Finally, the profitability of the routes is evaluated again and the local search heuristic is applied.

C. Logistics Service Provider

In order to increase the number of accepted transport requests, a carrier may outsource requests to a logistics service provider. The consideration has to be made whether a request is performed with the vehicle fleet of the carrier, outsourced to an LSP or rejected. In [13] a vehicle routing problem is studied, which allows subcontracting a part of the requests to external carriers. Their results show large cost savings by allowing subcontracting. The opportunity is also considered to outsource requests to external carriers in [17]. The objective is to minimise the sum of all costs, internal and external. Both the studies of [16] and [17] do not outsource to an LSP or reject. In our case, the pickup and delivery selection problem with an LSP (PDSPLSP) induces a fixed cost for each request outsourced to the LSP. The PDSPLSP checks if it is cheaper to perform the request using his own vehicle or to pay the LSP for the service. The third option is to reject a customer request. The optimal routes constructed for a carrier in the PDSPLSP are not necessarily the same as the optimal routes for the PDSP. In the PDSPLSP a carrier may decide to perform a request that remains non-served in the PDSP instead of an accepted request, even if it leads to a lower profit. This may be the case because the profit obtained from outsourcing a request, which is accepted in the PDSP, may be higher than the profit of outsourcing requests that are non-served in the PDSP. Thus, within the PDSPLSP the total profit of their own routes may be lower, but the profit obtained from the outsourced requests may be sufficiently high to balance this loss. This may lead to a higher total profit.

The insertion heuristic of the TSA algorithm is adapted to incorporate profits from outsourcing requests to an LSP. In the first step, the potential profit from outsourcing a request is calculated for all requests based on the difference between the revenue gained and the price paid to the LSP. In the next step, requests are added to routes the same way as in the insertion heuristic. However, it is not only checked whether the request is profitable, but also if it is more profitable than outsourcing a request to an LSP. When a request is inserted into a route of a carrier, it cannot be outsourced to an LSP. The INSERT and SWITCH operators are slightly modified. Each time a request is inserted into a route from the list of non-served requests, it is verified whether the profit is higher than the potential profit obtained when outsourcing the request. A request may be removed from a route, when a higher profit can be obtained from outsourcing the request to an LSP. Within the SHUFFLE(S) algorithm, the request is removed regardless of the potential profit to allow for extra diversification from the current local optimum. At the end of the algorithm, all

requests that remain non-served by the carrier are checked. If the fixed cost to outsource the request is less than the revenue gained from the customer, the request is outsourced to an LSP.

D. Alternative Revenue Model

The pricing of transport requests may be formulated in two ways: either the price is fixed resulting in a selection problem for the carrier to determine the profitable requests, or the pricing is performed according to the cost incurred to execute the request. Mostly the assumption is made that the price is fixed and the revenue depends on the distance between the pickup and delivery point of the customer. Hence, it is at the discretion of the carrier to select customers based on the potential revenue keeping in mind his limited resources.

In many real life cases, transport prices depend on the total number of units transported and not only on the distance travelled. The heuristic may include this variant. The only difference relates to the calculation of the proposed revenue in the input data. Therefore, instances where transport revenue depends on the number of units transported may be solved as well. This means that the assumption of a fixed revenue per kilometre is relaxed. Now the carrier has the opportunity to set his price according to the costs made to execute the transport requests. The objective is to try to accept as many customers as possible given the limited resources and later to determine the price that each customer has to pay to cover the total cost.

The problem at hand is a selection problem due to the fixed vehicle size and the limited number of working hours per day, but the objective is no longer to maximise the profit. The carrier aims to serve as many customers as possible during a single day. In a final step, the transport price for each customer is calculated based on the minimum price necessary to cover the cost made and a fixed profit margin. From the viewpoint of the carrier, it is assumed that there is no cost difference between the routing of an empty truck or a full truck. The cost of a truck is considered to depend only on the distance travelled. The extra fuel consumption for driving a loaded truck is considered to be marginal in the total cost of operating a truck per kilometre.

In literature, this type of selection problem, which maximises the number of accepted clients within a PDP, is not thoroughly studied. A variant of this selection problem, where resources are scarce, may be found in [18]. They propose a dynamic dial-a-ride problem in which the maximisation of the number of customers served is part of the objective function. The other two parts of the objective function are to optimise the level of service received by the customers and to minimise the total distance travelled by the vehicles.

The TSA-algorithm is split into two phases. First, the selection problem is solved to serve as many customers as possible. In the second phase, the selling price of transport requests is set. The first phase starts with the insertion heuristic. No revenue per request is given in the input data. The objective is to accept as many requests as possible in order to optimise the use of the available vehicles. Therefore, all requests are inserted as long as vehicle capacity is available and time windows are respected. When a request is inserted, it is placed into the route where it induces the lowest increase in the transport cost.

Furthermore, the local search operators are modified. The INSERT operator only checks whether an insertion is feasible, so neglecting the incurred cost of inserting a request. The incurred cost is the transport cost for travelling the additional distance to serve the extra client. The SWITCH, SHIFT and EXCHANGE operators remain the same. As there is no revenue, the transport cost before and after the change is compared instead of the profit. The improvement heuristic repeats the loop of local search operators as long as more requests are accepted (instead of improvement in profit as with the PDSP).

Finally, in the TSA algorithm the assessment criteria are changed to the number of accepted requests instead of the total profit over all routes. The DELETE operator is no longer used within the TSA algorithm, because the algorithm aims to insert as many requests as possible. The assessment criterion looks for an increase in the number of accepted requests; hence, removing a request reduces the probability of finding a better solution.

The second phase calculates the total cost for each vehicle, as the total distance travelled multiplied by the price per kilometre. Then, the marginal cost of each customer is determined. From the cost of a route with and without a customer, the difference of these costs is set as the marginal cost of that request. The total cost of a route is divided by the number of clients of that route based on the share of their marginal cost. Afterwards, a profit margin is added to the cost. This action gives the price asked to be paid by customers in order to execute their transport request.

VI. CONCLUSION

The operational planning decisions of a carrier consist of accepting transport requests of customers and constructing daily vehicle routes. Customers may have less-than-truckload requests to be transported between two specified locations, so these customers are consolidated into vehicle tours by the carrier. However, a carrier has a limited capacity within his own vehicle fleet. Therefore, he can only serve a selection of customers. Transport requests of customers are accepted only if they contribute to a higher total profit. A paired pickup and delivery selection problem is hardly investigated in literature. This paper tries to fill this gap. Two new local search operators, INSERT and SWITCH, are created to be able to handle the selection of transport requests. Together with the local search operators, SHIFT and EXCHANGE, an improvement heuristic is developed.

Next, a Tabu-embedded simulated annealing algorithm is proposed to solve the selection and routing problem. The TSA algorithm is able to further improve solutions found by the improvement heuristic. Although the TSA algorithm is not developed to solve PDP instances, tests on benchmark data by Li and Lim (2001) give good results. This indicates that the algorithm is able to construct short routes for the selected requests. Finally, alternative problem settings of the PDSP are considered. Different assumptions made in PDSP are relaxed. It is studied how these problem settings may be solved by using modified versions of the TSA algorithm. Implementation of the algorithms has been done in C++ and

Visual Studio. Experiments have been run on benchmark data sets. No experimental results are reported here.

ACKNOWLEDGEMENT

The present research has been supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office (Research Project COMEX, Combinatorial Optimization: Metaheuristics & Exact Methods).

REFERENCES

[1] S. Mitrovic-Minic (1998). "Pickup and delivery problem with time windows: a survey," Technical report TR1998-12, School of Computers Science, Simon Fraser University, Burnaby, BC, Canada. [Online]. Available: ftp://fas.sfu.ca/pub/cs/techreports/1998

[2] S. Mitrovic-Minic and G. Laporte, "The pickup and delivery problem with time windows and transhipment," *INFOR*, vol. 44, no. 3, 2006, pp. 217–227.

[3] S.N. Parragh, K.F. Doerner and R.F. Hartl, "A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations," *Journal für Betriebswirtschaft*, vol. 58, no. 2, 2008, pp. 81–117. http://dx.doi.org/10.1007/s11301-008-0036-4

[4] K. Ramaekers, A. Caris, G.K. Janssens and T. Maes, "Pickup and delivery selection with compulsory requests". In: P.J. Sequeria Gonçalves (Ed.), *Proceedings of the 11th Future Business Technology Conference (FUBUTEC'2015),* Lisbon, Portugal, 27–29 April 2015, pp. 28–33, (ISBN978-90-77381-88-5).

[5] C.-K. Ting and X.-L. Liao, "The selective pickup and delivery problem: formulation and a memetic algorithm," *International Journal of Production Economics*, vol. 141, no. 1, 2013, pp. 199–211. http://dx.doi.org/10.1016/j.ijpe.2012.06.009

[6] B. Verweij and K. Aardal, "The merchant subtour problem," *Mathematical Programming*, vol. 94, no. 2–3, 2003, pp. 295–322. http://dx.doi.org/10.1007/s10107-002-0321-2

[7] Y. Arda, Y. Crama and T. Pironet, "A profitable pickup and delivery problem with time windows," ORBEL 22 (22nd National Conference of the Belgian Operations Research Society). Booklet of abstracts, Brussels, 26–28 Jan. 2008, pp. 76–77. [Online]. Available: http://orbel22.hallot.net/ docs/BookletORBEL22.pdf

[8] T. Pironet, "The full truck-load profitable multi-vehicle one-to-one pick-up and delivery problem with hard time windows", Thesis DEA en Sciences de Gestion, HEC-Management School, Liège University, Belgium, 2007.

[9] L.F. Frantzeskakis and W.B. Powel, "A successive linear approximation procedure for stochastic, dynamic vehicle allocation problems," *Transportation Science*, vol. 24, no. 1, 1990, pp. 40–57. http://dx.doi.org/10.1287/trsc.24.1.40

[10] A.J. Kleywegt and J.D. Papastravou "Acceptance and dispatching policies for a distribution problem". *Transportation Science*, vol. 32, 1998, pp. 127–141. http://dx.doi.org/10.1287/trsc.32.2.127

[11] J. Schönberger, H. Kopfer and D. Mattfeld, "A combined approach to solve the pickup and delivery selection problem", in Leopold-Wildburger, U., Rendl, F. and Wascher, G. (Eds.), *Operations Research Proceedings* 2002, pp. 150–155.

[12] G. Desaulniers, J. Desrosiers, A. Erdmann, M.M. Solomon and F. Soumis, "VRP with pickup and delivery"., In: Toth P. and Vigo D. (Eds.), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, 2002, pp. 225–242. http://dx.doi.org/10.1137/1.9780898718515.ch9

[13] H. Li and A. Lim, "A metaheuristic for the pickup and delivery problem with time windows," in *Proceedings of the 13th IEEE International Conference on Tools and Artificial Intelligence*, Nov. 2001, Dallas TX, USA, pp. 160–167. http://dx.doi.org/10.1109/ictai.2001.974461

[14] J. Schönberger, *Operational Freight Carrier Planning*, Springer, Berlin, 2005.

[15] M. Ziebuhr and H. Kopfer, "The integrated operational transportation planning problems with compulsory requests," in Gonzalez-Ramirez, R.G., Schulte, F., Voβ, S., Ceroni Diaz J.A., *Proc. International Conference on Computational Logistics (ICCL2014), Lecture Notes in Computer Science 8760*, 2014, pp. 1–15, Springer. http://dx.doi.org/10.1007/978-3-319-11421-7_1

[16] H. Kopfer and X. Wang, "Combining vehicle routing with forwarding – extensions of the vehicle routing problem by different types of sub-contracting". *Journal of the Korean Institute of Industrial Engineers*, vol. 35 no. 1, 2009, pp. 1–14.

[17] M.-C. Bolduc, J. Renaud and F. Boctor, "A heuristic for the routing and carrier selection problem". *European Journal of Operational Research*, vol. 183, 2007, pp. 926–932. http://dx.doi.org/10.1016/j.ejor.2006.10.013

[18] A. Colorni and G. Righini, "Modeling and optimizing dynamic dial-a-ride problems," *International Transactions in Operational Research*, vol. 8, 2001, pp. 155–166. http://dx.doi.org/10.1111/1475-3995.00256

Katrien Ramaekers received degrees of *M.Sc.* in Business Engineering in 2002 and *Dr. sc. oec.* in Business Economics in 2007 from Hasselt University. From 2006 till 2008 she was a Researcher at the Transportation Research Institute of Hasselt University. From 2009 till 2014 she was a Post-doctoral Researcher at the Faculty of Business Economics, Hasselt University.

Currently she is a Professor of Operations Management and Logistics at Hasselt University (UHasselt) within the Faculty of Business Economics.

The focus of her work lied in simulation-optimisation approaches for inventory management. Other research has been developed in route decisions, more specifically for passengers, based on planned activities. Current research deals with simulation optimisation to apply flexible workforce scheduling both in a warehouse and in a production context. Furthermore she works on the allocation of costs and benefits for freight bundling networks in an intermodal transport.

Address: Faculty of Business Economics, Agoralaan 1, B-3590 Diepenbeek, Belgium.

E-mail: katrien.ramaekers@uhasselt.be

An Caris received degrees of *M. Sc* in Business Engineering in 2003 and *Dr. sc. oec.* in Business Economics in 2010 from Hasselt University. From 2011 till 2014 she was a Post-doctoral Researcher for the FWO (Fund for Scientific Research).

Currently she is a Professor of Operations Management and Logistics at Hasselt University (UHasselt) within the Faculty of Business Economics.

The focus of her work lied on the competitiveness of intermodal transport making use of inland navigation. Currently she works on other forms of collaboration in logistics. From a technical viewpoint, she also works on statistical analysis of algorithms for efficient parameter setting of meta-heuristic algorithms.

Address: Faculty of Business Economics, Agoralaan 1, B-3590 Diepenbeek, Belgium.

E-mail: an.caris@uhasselt.be

Tabitha Maes received degrees of *M. Sc.* in Business Engineering in 2009 and *Dr. sc. oec.* in Business Economics in 2013 from Hasselt University.

Since 2013 she has been employed as a Supply Chain Engineer at Scania Parts Logistics in Opglabbeek, Belgium, which is the world-wide distribution centre for Scania parts. She is currently involved in the world-wide roll-out of a new planning system in their network.

Her doctoral research dealt with logistics decisions by carriers, more specifically in the decision making of selecting transport requests.

Address: Scania Parts Logistics, Bedrijfsstraat 1208, B-3660 Opglabbeek, Belgium.

E-mail: tabitha.maes@scania.com

Gerrit K. Janssens received degrees of *M. Sc.* in Business Engineering from the University of Antwerp (RUCA), Belgium, *M. Sc.* in Computer Science from the University of Ghent (RUG), Belgium, and *Dr. sc. ing.* in Computer Science in 1989 from the Free University of Brussels (VUB), Belgium. After some years of work at General Motors Continental, Antwerp, he joined the University of Antwerp.

Currently he is a Professor of Operations Management and Logistics at Hasselt University (UHasselt) within the Faculty of Business Economics.

He was the President of the Belgian Operations Research Society (ORBEL) in 2006–2007. During the last twenty-five years he has been several times visiting universities in South-East Asia and Africa. His main research interests include the development and application of operations research models in production and distribution logistics.

Address: Faculty of Business Economics, Agoralaan 1, B-3590 Diepenbeek, Belgium.

E-mail: gerrit.janssens@uhasselt.be