Distributing Fibre Boards: A Practical Application of the Heterogeneous Fleet Vehicle Routing Problem with Time Windows and Three-Dimensional Loading Constraints

Shannon Pace[1], Ayad Turky[2], I. Moser[3], and Aldeida Aleti[4]

[1] Swinburne University of Technology, Melbourne, Victoria, Australia
`space@swin.edu.au`
[2] Swinburne University of Technology, Melbourne, Victoria, Australia
`aturky@swin.edu.au`
[3] Swinburne University of Technology, Melbourne, Victoria, Australia
`imoser@swin.edu.au`
[4] Monash University, Melbourne, Victoria, Australia
`aldeida.aleti@monash.edu`

Abstract

The Heterogeneous Fleet Capacitated Vehicle Routing Problem with Time Windows and Three-Dimensional Loading Constraints (3L-HFCVRPTW) combines the aspects of 3D loading, heterogeneous transport with capacity constraints and time windows for deliveries. It is the first formulation that comprises all these aspects and takes its inspiration from a practical problem of distributing daily fibre board deliveries faced by our industry partner. Given the shape of the goods to transport, the delivery vehicles are customised and their loading constraints take a specialised form. This study introduces the problem and its constraints as well as a specialised procedure for loading the boards. The loading module can be called during or after the route optimisation. In this initial work, we apply simple local search procedures to the routing problem to two data sets obtained from our industry partner and subsequently employ the loading module to place the deliveries on the vehicles. Simulated Annealing outperforms Iterated Local Search, suggesting that the routing problem is multimodal, and operators that shift deliveries between routes appear most beneficial.

Keywords: Vehicle Routing Problem, Time Windows, 3-Dimensional Loading Constraints, Local Search, Simulated Annealing

1 Introduction

The VRP has been widely researched in the optimisation literature over the last 40 years. More recently, vehicle routing has been combined with container loading [3], where the objective is to find the shortest route given constraints posed by the 3-dimensional shapes of the goods

to deliver. Since 2006, a number of studies have addressed this problem with the inclusion of capacity constraints, loading constraints or time windows. None of the approaches to date include all three aspects.

Generally, the 3D container loading problem is formulated as a placement of rectangular objects within a rectangular space. Recently, a problem of distributing wooden boards has been introduced in a study by Doerner et al. [6]. The boards have to be placed on pallets and are modular in the sense that they cover one or two pallets at a time. The problem that is faced by our industry partner is potentially more complex, as thousands of sizes of fibre boards and laminates are kept in stock. Fortunately, the boards delivered on a daily basis are most often comprised of four sizes only. The trucks' loading areas are flat surfaces with slots for support poles which keep the stacks of boards in place. The deliveries of individual customers are first bundled into one or more packs which can be stacked. In general, larger boards have to be placed under smaller boards. A strict LIFO order is mandatory: All packs belonging to a single customer have to be on top of their respective stacks when the truck arrives at the customer's site.

The loading optimiser also has to consider back-to-front and side-to-side balance constraints to maintain road safety throughout the each route. A customised fleet of three types of vehicles and restricted drop-off times with some customers make further demands on the optimisation algorithm.

Our industry partner wishes to maximise the number of deliveries made during a single day. Customers who order by a certain time are guaranteed next-day delivery. If the available fleet cannot distribute the deliveries within the ten daily working hours allowed for each driver, external contractors have to be engaged.

This paper describes and formalises the problem as well as introduces a loading algorithm based on a depth-first tree search. The loading module is tested after a route optimisation procedure implemented as a simulated annealing (SA) approach, which is compared to an alternative based on iterative local search (ILS), where both approaches use varied combinations of local search operators. Feasible results are compared in terms of the overall delivery time for all items to be distributed on the day.

2   Existing Work

The Capacitated Vehicle Routing Problem (CVRP) is a shortest path problem that has been the topic of a considerable number of studies, the most relevant of which have been included in a comprehensive overview by Toth and Vigo [16] in 2002. Professionals in industry are most often faced with both the CVRP and the secondary problem of placing delivery items in the vehicle, which can be seen as a form of the classical combinatorial problem of bin packing. Cagan, Shimada and Yin [4] have presented a survey of algorithms for three-dimensional packing problems. Since the problem is NP-hard, it is often solved to acceptable quality using a heuristic. The most sophisticated exact algorithm known today [9] is able to solve problems with up to 135 customer drop-offs.

The integrated problem of CVRP and three-dimensional load construction (3L-CVRP) has only been considered by researchers very recently. In all cases, the problem is described in terms of fitting three-dimensional rectangular boxes into a cubic space. The seminal work was presented by Gendreau et al. [10] in 2006. Their approach adapts the bottom left algorithm [1] and the touching perimeter algorithm [12] to the packing problem using a taboo list.

Solutions proposed to the 3L-CVRP include the approach by Tarantilis, Zachariadis and Kiranoudis [15], who devised six heuristics for the placement of the rectangular boxes in a

vehicle, an interesting approach by Duhamel, Lacomme and Toussaint [7] who proposed an algorithm that combines all tours, perturbs them and splits them up into routes again, as well as an Ant Colony Optimisation approach by Fuellerer et al. [8]. Bortfeldt [2] shifts orders between routes with the goal of minimising the number of vehicles.

Many of these approaches [10, 18, 6, 17] create their initial solutions based on the 'savings algorithm' by Clarke and Wright [5], which was adapted for ACO by Reimann, Doerner and Hartl [14], a method that also made part of Fuellerer et al.'s work [8]. Toth and Vigo [16] developed Clarke and Wright's algorithm into a multi-start randomised savings procedure.

All approaches to 3L-CVRP known to date include a local search procedure, the most sophisticated use a combination of relocations of orders within and between routes as well as swaps within and between routes [15, 2], and taboo lists are commonly used to avoid repeating local search moves [10, 15, 8, 6] Moura and Oliveira [13] also consider time windows, whereas Wei, Zhang and Lim [18] optimise the problem with a heterogeneous fleet; in the combination of 3L-HFCVRPTW, the problem does not seem to have been addressed as yet.

One of the two known loading problems with similar loading constraints as in the current work was published by Doerner et al. [6], where a company distributes wooden chipboards to manufacturers. This multi-pile (MP) loading problem has to accommodate only four types of boards which fit on standard pallets. The vehicles can fit one pallet's width and three pallets along the length of the truck. The weight capacity is not needed since the boards are not heavy enough to reach the bounds at full volume, and the width can be ignored because all boards are at most as wide as a pallet. Items can span more than one pile. For a feasibility estimation the authors propose a recursive algorithm that determines the maximal height needed by a combination of the deliveries of a pair of customers. The VRP is solved using the savings algorithm [5] with a taboo list and a 4-opt local search. A second formulation uses the savings heuristic as part of an ACO solution construction approach in which the pheromone values describe the usefulness of customer i following customer j in a route. A penalty value is added to the objective function for excess height of the piles. The authors observe that larger problems with more than 75 deliveries are better solved using ACO.

Tricoire et al. [17] further investigated the MP loading problem and devised an exact algorithm for constructing a load for a single route. A variable neighbourhood search (VNS) is applied after an initial solution has been constructed using the savings algorithm [5]. One of the perturbations is the exchange of segments of routes while preserving the order, another is 2-opt. An exact branch-and-cut algorithm is also applied to the problem. A comparison of the VNS with the approaches of Doerner et al. [6] concluded in favour of the VNS, which finds better solutions in 90% of the cases. Running the branch-and-cut algorithm for 24 hours did not lead to significant improvements in the result quality.

3   Description of 3L-HFCVRPTW

The primary aim of the paper is to formalise a variation of the 3L-HFCVRPTW that is faced in practice by our Australian industry partner. The subsidiary of a large building company manufactures and distributes particle boards, medium density fibre boards and high pressure laminates. These stock keeping units (SKUs) exist in many colours and sizes and are used as building material for doors, kitchen cabinets and similar. The company has a distribution center (DC) in major Australian cities (Melbourne, Adelaide, Brisbane). This work captures the distribution problem faced by the Melbourne DC (MDC).

The MDC is equipped with three types of custom-made delivery trucks. Customers are guaranteed next-day delivery if they order by 4pm, after which the deliveries are planned and

the trucks loaded for the next day. Ideally, the company would not have to hire external delivery capacity, although this could not always be avoided in the past. Truck drivers must not work more than 10 hours a day. If they return to the MDC in time, they can take another load and service another route if they can do so without exceeding their working hours.

Customer orders can include one or more boards of the same size. All boards of the same size, regardless of type, are bundled into a pack to be delivered to the customer. If a customer orders different sizes, several packs have to be created with supports and cover sheets, then added in LIFO order to a suitable stack on the truck. The load has to be balanced back-to-front (60% – 40%) and side-to-side (70% – 30%).

The problem at hand can formally be described as a complete graph G (V, E) where V represents a set of customers $V = \{0, ..., n\}$, 0 denotes the MDC and $\{1, ..., n\}$ represent customers. The vertices are connected by edges $E = \{e_{ij} : i \neq j \ and \ i, j \in V\}$.

Objective Function: Each edge is associate with a travel distance c_{ij} expressed in minutes. Melbourne traffic is heavy on certain roads and fluctuates greatly during the day, hence distances do not reflect the problem accurately. The assumption $c_{ij} = c_{ji}$ is a simplification made for this first formulation. Eq. 1 expresses the summation of the cost; e_{ij} takes the value 0 if the edge between customers i and j is not used, 1 otherwise. Position 0 denotes the MDC and connects to twice as many customers as there are routes. Waiting times δ_i only apply when the truck arrives before the start of the time window of customer i.

$$f(x) = \sum_{i=0}^{n} \sum_{j=0, j\neq i}^{n} e_{ij} c_{ij} + \sum_{i=1}^{n} \delta_i \tag{1}$$

3.1   Routing Constraints

The routes must start and end at the MDC and obey possible time windows (including the limit on a driver's total working hours).

Service times: Depending on the availability of a forklift on site, and the number of packs to be delivered, customer drop-offs may take between 5 and 25 minutes, considered as the service time s_i. This is relevant for the satisfaction of constraints but can be ignored as irrelevant in the objective function.

Time windows: The MDC carries out all its daily deliveries between 7:00 and 17:00 hours and all customers must have been serviced and the trucks returned within this time interval. For most customers, the delivery time window is identical with this. In some cases, the customer may have the ability to receive deliveries only at particular times. To accommodate these restrictions, we have to assume time windows as follows. A customer k whose time window $tw_k = t_k^e - t_k^s$ is shorter than the delivery interval of the MDC must be serviced according to the constraint in Eq. 2, which expresses that all travel $c_{i,i+1}$ and service times s_i of the customers $\{0...k-1\}$ in route r_x combined have to be greater than or equal to the start time of customer k's time windows start time t_k^s and smaller than the end time t_k^e reduced by the service time s_k of customer k. It is assumed that the load has to be unloaded before the end of the time window, and that an early arrival incurs a waiting time δ which adds to the objective value of the solution.

$$t_k^s - \delta \leq \sum_{i=0, i \in r_x}^{k-1} c_{i,i+1} + s_i \geq t_k^e - s_k \tag{2}$$

3.2   Loading Constraints

The loading constraints are multifaceted: The boards have to be stacked so that each board is supported, the load is balanced after each drop-off and the overall capacity of the truck is not exceeded.

Fleet: The MDC is equipped with a fleet of three types of specialised trucks – eight 8tn trucks, nine 12tn trucks and one 20tn semitrailer, each with a flat loading surface and spaces for dividers, which keep stacks of packs of boards apart. Each stack must be stabilised with a dividing pole in its front, which sets boundaries to the possible placements of the stacks of the boards. The trucks can be unloaded from both sides, which are covered by tarpaulins during transit.

Weight: Each customer i requires a supply of m_{ik} items, whose total weight d_i is known. The deliveries carried by a truck must not exceed its weight limit.

Layout: Each item has a length l_{ik}, a width w_{ik} and a height (or thickness) $3mm <= h_{ik} >= 25mm$. In general, the weight constraint is tighter than the volume constraint, but naturally there are combinations of board sizes that cannot be accommodated on a truck. Split deliveries are not permitted. The SKUs that form part of a customer's deliveries have to be divided into stacks of a suitable size. Naturally, the widths and lengths of the SKUs in a stack must not reach beyond the loading surface of the truck, and the layout of the stacks must permit the addition of dividing poles between all stacks. Dividers can be inserted into slots on the loading surface.

Stacks are formed by grouping SKUs of similar sizes. Variations in sizes within the same stack are possible, subject to the following rules:

1. The first board in a stack defines the base size of the stack.

2. The subsequent boards can be larger than the base size if they

    (a) do not exceed the dimensions of the truck;

    (b) do not exceed the base size by more than half in either direction.

3. Boards higher in the stack can be smaller than the base size if

    (a) they are the topmost boards, or covered by smaller boards only;

    (b) they are covered by two-thirds of the length and/or width of a larger board.

A customer's SKUs are grouped into *packs* before being added to a stack. To speed up the drop-offs, a minimal number of packs per customer is desirable. At most, there are as many packs per customer as there are stacks. The packs are stacked strictly in LIFO order.

Load Balance: A *balanced load* ought to be maintained after each drop-off. Ideally, the rear axle (rear half of the loading area) carries 60% of the load weight. Loading the rear half of the truck with up to 60% of the maximum weight capacity of the truck before adding any weight to the front half is considered a tight constraint which must not be violated. A 70%-30% left/right balance is considered a reasonable limit for load distribution, but this is not always maintained. If a customer's SKUs are of a size that permits them to occupy only one stack, they are bundled into a single pack to save time.

4   Algorithmic Approaches

4.1   Route Optimisation

Since the aim is to find an acceptable solution that can be implemented in practice while further improvements are being sought, Iterated Local Search (ILS) and Simulated Annealing (SA) are applied in the first instance to 3L-HFCVRPTW. Both ILS and SA are simple local search methods which start from an initial solution and then iteratively move from one solution to a neighbouring one.

Initially, an empty route is created with a truck picked from a list ordered by capacity (smallest first) and customer deliveries are chosen uniformly randomly and assigned to this route. This is repeated as long as the weight constraint of the truck is not exceeded. Assignments are also rejected if the time window constraint is violated. When no further customer can be added, a new route is created. This process is repeated until all customers have been assigned to a route.

Neighbourhood Operators: The ILS and SA approaches share the same neighbourhood operators, which are 2-opt, OR-opt and Shift/Mutate:

- 2-opt chooses two deliveries randomly within a route and transposes them.

- OR-opt selects a delivery from a route and moves it to a different location in the same route.

- Shift selects a random customer delivery to move to a different route.

The algorithms are comprised of one 'large' move (Shift) and one 'small' one (either 2-opt or OR-opt) at a time. The feasibility of the solution is maintained throughout the neighbourhood search process.

4.1.1   Iterated Local Search

The local search algorithm used here is simple ILS which chooses the orders to reassign uniformly randomly and accepts the move if it entails a fitness improvement. It uses one attempt per iteration. The stopping criterion is a predefined number of iterations.

4.1.2   Simulated Annealing

SA is one of the predominant heuristic approaches used to address the VRP problem. A recent application by Kokubugata and Kawashima [11] addresses a similar problem as the one faced by our industry partner, city logistics. In this case, SA starts from an initial solution. SA always

accepts an improving local search move but applies a probability of acceptance to deteriorating moves. The probability P of accepting a worse solution is calculated using Eq. 3

$$P = exp^{\frac{-(f(x')-f(x))}{t}} \tag{3}$$

where $f(x)$ is the fitness of the current solution and $f(x')$ the solution after the proposed change and t is the current temperature. The pseudocode of SA is shown in Algorithm 1.

Algorithm 1: Simulated Annealing for Route Optimisation

---
1 $x \leftarrow$ random initial solution ;
2 $t_i \leftarrow$ initial temperature ;
3 $t_k \leftarrow$ final temperature ;
4 $r \leftarrow$ cooling rate ;
5 **while** $t_i > t_k$ **do**
6 $\quad x' \leftarrow$ apply neighbourhood operator to x;
7 $\quad$ calculate $f(x')$;
8 $\quad$ **if** $random[0,1] < exp^{\frac{-(f(x')-f(x))}{t}}$ **then**
9 $\quad\quad$ $\lfloor$ $x' \leftarrow x$;
10 $\quad t_{i+1} \leftarrow r * t_i$;
11 $\quad i \leftarrow i + 1$;

4.2   Loading Heuristic

A specialised heuristic had to be devised to accommodate boards in the customised vehicles. The loading module receives a permutation of customer orders prescribed by the outcome of the routing algorithm. Each customer's order consists of boards of different lengths and widths (the thicknesses are ignored at this stage). The boards are grouped into batches according to length and width.

Once all sizes have been determined, a preliminary layout on the bottom of the truck including the number of 'rows' of stacks is decided. Because all parts of a customer delivery have to be accessible from either side of the truck, a row can have one or two stacks abreast. A rough estimate of the expected stack weights decides where the stacks are placed on the truck.

A depth-first search creates a tree structure that places batches into packs which are placed on stacks. The search proceeds in layers where one layer comprises the packs needed to include one customer's items. If a pack placement leads to an invalid solution given the constraints outlined in section 3.2 and the heights of the stacks, it is removed and possibly re-packed before another placement on stacks is attempted. If none of the possible placements succeeds, the algorithm backtracks to remove further customer layers before rebuilding the layers again. The recursive backtracking procedure is exhaustive and only viable due to the relative homogeneity of the board sizes and the fact that the choices of alternative packs are limited by the number of stacks.

When the stacks have been created, the algorithm checks whether the dimensions of the stacks actually allow them to be placed on the loading surface considering the positions of the dividers.

Algorithm 2: Algorithm for loading module

Data: open customer orders o, truck layout T
Result: order configuration on truck C

1 $o \leftarrow \varnothing$;   // Closed customer orders
2 $e \leftarrow \varnothing$;   // Stack of visited tree nodes
3 $s \leftarrow \texttt{EstimateRequiredStacks}(o, T)$;
4 $C \leftarrow \texttt{InitialiseConfiguration}(s, T)$;
5 **while** $\|o\| > 0$ **do**
6   **if** $\|p\| > 0$ **then**
7        $C \leftarrow \texttt{AddPack}(C, \texttt{PopLeft}(\texttt{Peek}(e)))$;   // Advance.
8        $a \leftarrow \texttt{PopLeft}(o)$;
9        $c \leftarrow \texttt{Push}(c, a)$;
10       $b \leftarrow \texttt{CreateBatches}(a)$;
11       $p \leftarrow \texttt{GenerateMultichoosePermutations}(s, b)$;
12       $p \leftarrow \texttt{FilterImpossiblePermutations}(C, p)$;
13       $p \leftarrow \texttt{SortPermutationsByHeuristicDesc}(C, p)$;
14       $e \leftarrow \texttt{Push}(e, p)$;
15   **if** $\|e\| > 0$ **then**
16       $C \leftarrow \texttt{RemoveLastPack}(C)$;   // Backtrack.
17       $\texttt{PopRight}(e)$;
18       $o \leftarrow \texttt{Push}(o, \texttt{PopRight}(c))$;
19   **else**
20       $\texttt{Abort}()$;   // No complete solution could be found.

---

5   Experimental Settings

Two initial data sets (L1 and L2) were collected from order sheets obtained from our industry partner, each of them describing the orders of a single day. One data set contained 158 customer orders, the other 130. The customer locations were obtained in the form of GPS coordinates. The sizes ordered were 1800x3600mm, 1200x3600mm, 1800x2400mm and 1200x2400mm with thicknesses between 3mm and 25mm. About 20 customers in each set had time windows of 1hr – 7hrs, all others could be visited at any time during the day.

The data sets were optimised using ILS and SA followed by the load optimisation. All trials were repeated 30 times. The algorithms used a single initial solution, allowing an equal number of function evaluations for the improvements. At each iteration, one of the operators is chosen with equal probability. SA makes the change permanent depending on the temperature-based probability. ILS applies only improving changes.

SA starts from an initial temperature of $t_k = 100$ and applies a cooling rate of $r = 0.99$ which is applied after each iteration. These settings were found optimal during preliminary trials.

6   Results and Discussion

The combinations of 2-opt and Shift as well as OR-opt and Shift were applied with both algorithms, denoted ILS 2-opt/Shift, ILS OR-opt/Shift, SA 2-opt/Shift and SA OR-opt/Shift.

Table 1: The best and mean fitnesses and standard deviations of the 30 runs of ILS and SA with two combinations of neighbourhood operators and the number of changes the algorithms made to the solution.

	Operator	L1				L2			
		Best	Avg	Stdev	# Changes	Best	Avg	Stdev	# Changes
ILS	2-opt/Shift	3130	3435	180	1263	2213	2530	193	1285
	OR-opt/Shift	3149	3492	186	1047	2212	2472	153	1091
SA	2-opt/Shift	3076	**3340**	**159**	2396	**1974**	**2291**	**152**	2666
	OR-opt/Shift	**2964**	3343	196	2085	1989	2331	160	1995

The results are presented in Table 1. The best results are shown in bold font.

The results are shown in total delivery time in minutes. They seem to indicate that the search landscape is multimodal and too complex to solve optimally with a simple local search algorithm. SA clearly outperforms local search by admitting deteriorating moves. Given an equal number of function evaluations, SA expectedly makes more changes than ILS. Transposition (2-opt) moves appear more beneficial than repositioning moves. Intuitively, repositioning a single order may not lead to a fitness improvement if the recipient of the neighbouring delivery is far from the repositioned order, but would ideally be placed elsewhere in the route. A transposition (2-opt) move can overcome such a situation, which makes it less prone to entrapment in local optima.

However, in the case of the larger problem L1, the best solution was found by the OR-opt move. This may indicate that the OR-opt move can still have benefits when the solution is close to the - local or global - optimum already. To take advantage of this move, however, it has to be integrated in a meaningful way with the 2-opt and Shift moves. When used on its own, the high standard deviation shows that its quality fluctuates greatly, again pointing towards entrapment in local optima.

7   Conclusion and Further Work

The 3L-HFCVRPTW formulation presented here is based on a practical problem faced in industry with very specific loading constraints. The initial optimisation approach introduced here forms the basis for possible further optimisation. The 2-opt operator appears most efficient at intra-route optimisation. Inter-route optimisation relies on the Shift operator which repositions a single delivery in a different route. More sophisticate operators have to be devised which may use transpositions. It may be beneficial to use a - perhaps distance-based - heuristic to choose the deliveries to swap/reposition.

The algorithm did not check whether a route solution would lead to a feasible load before the end of each trial. Every optimum found turned out to be feasible when submitted to the loading module. This is a positive surprise as it leaves much opportunity for route optimisation. Whereas the loading constraints are very tight in theory, the practical data sets appear to leave plenty of opportunities for optimisation.

In this initial approach we did not attempt to perform restarts after the end of the local optimisation performed by ILS/SA. In the future, we may encounter infeasible solutions and it may be helpful to have a second best solution at hand for this case. Therefore, restarts are not only an opportunity to escape local optima but also a way to enforce robustness with an algorithm that is to be used in practice.

References

[1] B. Baker, E. Coffman, Jr., and R. Rivest. Orthogonal packings in two dimensions. *SIAM Journal on Computing*, 9(4):846–855, 1980.

[2] Andreas Bortfeldt. A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, 39(9):2248 – 2257, 2012.

[3] Andreas Bortfeldt and Gerhard Wäscher. Constraints in container loading - A state-of-the-art review. *European Journal of Operational Research*, 229(1):1–20, 2013.

[4] J. Cagan, K. Shimada, and S. Yin. A survey of computational approaches to three-dimensional layout problems. *Computer-Aided Design*, 34(8):597 – 611, 2002.

[5] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

[6] Karl F. Doerner, Guenther Fuellerer, Manfred Gronalt, Richard F. Hartl, and Manuel Iori. Meta-heuristics for the vehicle routing problem with loading constraints. *Networks*, 49:294–307, 2007.

[7] Christophe Duhamel, Philippe Lacomme, Alain Quilliot, and Hélène Toussaint. A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research*, 38(3):617 – 640, 2011.

[8] Guenther Fuellerer, Karl F. Doerner, Richard F. Hartl, and Manuel Iori. Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, 201(3):751 – 759, 2010.

[9] Ricardo Fukasawa, Jens Lysgaard, Marcus Poggi de Arago, Marcelo Reis, Eduardo Uchoa, and RenatoF. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. In Daniel Bienstock and George Nemhauser, editors, *Integer Programming and Combinatorial Optimization*, volume 3064 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin Heidelberg, 2004.

[10] Michel Gendreau, Manuel Iori, Gilbert Laporte, and Silvaro Martello. A tabu search algorithm for a routing and container loading problem. *Transportation Science*, 40(3):342–350, 2006.

[11] Hisafumi Kokubugata and Hironao Kawashima. Application of simulated annealing to routing problems in city logistics. *Simulated Annealing*, page 420, 2008.

[12] Andrea Lodi, Silvano Martello, and Daniele Vigo. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 11(4):345–357, 1999.

[13] Ana Moura and José Fernando Oliveira. An integrated approach to the vehicle routing and container loading problems. *OR Spectrum*, 31(4):775–800, 2009.

[14] Marc Reimann, Karl Doerner, and Richard F Hartl. D-ants: Savings based ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, 31(4):563 – 591, 2004.

[15] C.D. Tarantilis, E.E. Zachariadis, and C.T. Kiranoudis. A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *Intelligent Transportation Systems, IEEE Transactions on*, 10(2):255–271, June 2009.

[16] Paolo Toth and Daniele Vigo. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2002.

[17] Fabien Tricoire, KarlF. Doerner, RichardF. Hartl, and Manuel Iori. Heuristic and exact algorithms for the multi-pile vehicle routing problem. *OR Spectrum*, 33(4):931–959, 2011.

[18] Lijun Wei, Zhenzhen Zhang, and A. Lim. An adaptive variable neighborhood search for a heterogeneous fleet vehicle routing problem with three-dimensional loading constraints. *Computational Intelligence Magazine, IEEE*, 9(4):18–30, Nov 2014.