The Vehicle Routing Problem with Simultaneous Pick-Ups and Deliveries and Two-Dimensional Loading Constraints

Emmanouil E. Zachariadis[a], Christos D. Tarantilis[a], Chris T. Kiranoudis[b]*

[a] *Department of Management Science & Technology*, *Athens University of Economics and Business*

[b] *School of Chemical Engineering, National Technical University of Athens*

[*] Corresponding Author (email: ezach@aueb.gr)

Abstract

We introduce and solve the Vehicle Routing Problem with Simultaneous Pick-ups and Deliveries and Two-Dimensional Loading Constraints (2L-SPD). The 2L-SPD model covers cases where customers raise delivery and pick-up requests for transporting non-stackable rectangular items. 2L-SPD belongs to the class of composite routing-packing optimization problems. However, it is the first such problem to consider bi-directional material flows dictated in practice by reverse logistics policies. The aspect of simultaneously satisfying deliveries and pick-ups has a major impact on the underlying loading constraints: feasible loading patterns must be identified for every arc traveled in the routing plan. This implies that 2L-SPD generalizes previous routing problem variants with two-dimensional loading constraints which call for one feasible loading per route. From a managerial perspective, the simultaneous service of deliveries and pick-ups may bring substantial cost-savings, but the generalized loading constraints are very hard to tackle in reasonable computational times. To this end, we propose an optimization framework which employs memorization techniques designed for the 2L-SPD model, to accelerate the solution methodology. To assess the performance of our routing and packing algorithmic components, we have solved the Vehicle Routing Problem with Simultaneous Pick-Up and Deliveries (VRPSPD) and the Vehicle routing Problem with Two-Dimensional Constraints (2L-CVRP). Computational results are also reported on newly constructed 2L-SPD benchmark problems. Apart from the basic 2L-SPD version, we introduce the 2L-SPD with LIFO constraints which prohibit item rearrangements along the routes. Computational experiments are conducted to understand the impact of the LIFO constraints on the routing plans obtained.

Keywords: *Logistics*; *Vehicle Routing*; *Loading Constraints*; *Simultaneous pick-ups and deliveries*; *Heuristics*.

1. Introduction

In the last years, advances both in optimization methodologies and computer systems allow researchers and practitioners to examine practical optimization problems which in the past were thought to be too complex to handle. One such research stream that has emerged in the logistics optimization literature is devoted to the analysis of problems which are aimed at effectively dispatching a fleet of vehicles and at the same time, ensuring that the transported items can be feasibly loaded into these vehicles. The problem introduced in the present study belongs to this class of integrated vehicle routing and loading problems. Briefly, the presented model, referred to as *the Vehicle Routing Problem with Simultaneous Pick-Ups and Deliveries and Two Dimensional Loading Constraints* (2L-SPD) calls for the generation of the optimal routes to fully satisfy the demand raised by a

set of customers. The demand of each customer consists of two transportation requests: the first one is associated with a set of items that must be transported from the warehouse to the customer location, whereas the second request is associated with a set of items that must be transported from the customer location to the central warehouse. The items that are transported from and to the warehouse are considered rectangular and not stackable. Thus, 2L-SPD is aimed at generating feasible, two-dimensional, orthogonal loading patterns for the transported items carried by the produced route set.

The main innovative feature of the examined model, compared to previously introduced vehicle routing variants with loading constraints, lies in the fact that vehicles offer simultaneous pick-up and delivery service. This implies that the item sets carried along a route change drastically: delivery items are unloaded and additional pick-up items are loaded onto the vehicle. Thus, the loading feasibility must be examined for every arc traveled by the routes. On the contrary, previously introduced delivery models assume that the size of the item set carried by a vehicle monotonically decreases, so that the loading feasibility has to be tested, only when the corresponding vehicle leaves the warehouse fully loaded. As a result, previously examined delivery models with two-dimensional loading constraints can be regarded as a special case of 2L-SPD, when for all customers, the pick-up requests are set to an empty item set.

Regarding the routing characteristics, 2L-SPD is a generalization of the vehicle routing problem with simultaneous pick-ups and deliveries (VRPSPD) which calls for the optimal routes that simultaneously offer pick-up and delivery service, under one-dimensional loading constraints (Dell'Amico et al., 2006; Subramanian et al. 2013a). Analogously, VRPSPD is a generalization of the basic version of the vehicle routing problem (VRP) which is aimed at producing the optimal delivery route set subject to one-dimensional capacity constraints.

As already stated, 2L-SPD belongs to the integrated vehicle routing and multi-dimensional packing problems which jointly call for the optimal route planning and feasible packing structures for the transported goods. The first such problem has been introduced by Iori et al. (2007) who examine a vehicle routing extension with two-dimensional loading constraints: vehicles are considered to deliver rectangular items (boxes, pallets) which are not stackable. This problem is referred to as the vehicle routing problem with two-dimensional loading constraints (2L-CVRP). Under 2L-CVRP, the minimum cost set of routes must be generated for the vehicle fleet. For each of these routes, a feasible orthogonal two-dimensional packing must be determined for the transported items. The authors present a branch-and-cut method for dealing with small-scale problems (up to 25 customers and 91 boxes). To solve larger-scale instances, researchers have proposed various metaheuristic solution strategies: A tabu search methodology has been developed by Gendreau et al. (2008). Zachariadis et al. (2009) have proposed a tabu search and guided local search hybridization for the routing aspects and a bundle of packing heuristics for the loading requirements. Fuellerer et al. (2009) have developed an ant colony optimization approach. Another tabu search-guided local search hybrid has been proposed by Leung et al. (2011). Strodl et al. (2010) have proposed a 2L-CVRP solution method emphasizing on the development of efficient data structures for storing obtained loading feasibility information. More recently, Duhamel et al. (2011) have solved the 2L-CVRP by a greedy randomized adaptive search (GRASP) and evolutionary local search (ELS) solution approach, while Zachariadis et al. (2013) have proposed a simple-structured local search methodology. The most recent works on the 2L-CVRP are due to Dominguez et al. (2014) and Wei et al. (2015) who introduce a Variable Neighborhood Search method. An additional routing model with two-dimensional loading constraints has been introduced by Malapert et al. (2008). The authors present a pick-up and delivery

model which assumes that non-stackable rectangular items have to be transported between pairs of service locations. An additional class of integrated routing-packing models considers three-dimensional loading constraints. This model category is applicable for logistics applications where the transported boxes can be stacked one on top of the other. The first such study is due to Gendreau et al. (2006). Their work introduces the vehicle routing problem with three-dimensional loading constraints (3L-CVRP) which generalizes 2L-CVRP by calling for feasible, three-dimensional loading arrangements. Additional requirements met in practice are examined: fragility constraints, stability rules for the transported cargo and easy unloading operations. Several metaheuristic developments have been proposed for the 3L-CVRP (Tarantilis et al. 2009; Fuellerer et al., 2010; Ruan et al. 2011; Zhu et al. 2012; Bortfeldt, 2012). A relevant model is due to Männel and Bortfeldt (2013). The latter work introduces a pickup and delivery problem where three-dimensional and stackable items are transported between customer locations. For a detailed list of vehicle routing models which explicitly deal with loading constraints, the interested reader is referred to the reviews of Iori and Martello (2010), Iori et al. (2013) and Perboli et al. (2014).

The purpose of the present paper is to formally introduce the 2L-SPD model. An efficient solution approach is developed and presented for the 2L-SPD. The proposed solution approach consists of two algorithmic components: one for the routing and one for the packing aspects. Both components are based on our algorithm presented for the 2L-CVRP (Zachariadis et al., 2013). However, they are extended to provide higher-quality solutions. In addition, we present a new original framework which combines the individual routing and packing components for efficiently dealing with the special requirements of the 2L-SPD model. We also present feasibility memory structures that have been specially designed for the 2L-SPD and drastically accelerate the search process. The overall 2L-SPD solution approach is a robust optimization methodology efficiently dealing with instances of hundreds of customers and items.

In addition to the basic 2L-SPD model, we introduce the 2L-SPD with LIFO constraints. Under the LIFO variant, item rearrangement along the routes is not allowed, so that the loading requirements become much tighter leading to lower area utilization. To assess the effectiveness of both the routing and packing ingredients of our algorithm, computational results are reported on well studied VRPSPD and 2L-CVRP benchmark instances. Then, computational results are reported on newly constructed 2L-SPD test cases both for the basic 2L-SPD, as well as the LIFO constrained variant.

The remainder of the present paper is as follows: Section 2 presents in detail the examined problem and discusses its applicability for practical logistics operations. Section 3 describes the proposed 2L-SPD local search solution approach. This master local search algorithm makes use of two loading feasibility examination components which are described in Sections 4 and 5. Then, Section 6 provides the necessary methodological modifications for tackling the LIFO version of the 2L-SPD model. In Section 7, extensive computational results are reported for the VRPSPD, 2L-CVRP and 2L-SPD models. In addition, comparisons are made between the obtained results and the ones of previously published methodologies. Finally, Section 8 concludes the paper.

2. The 2L-SPD Model

In this Section, we present a formal description of the 2L-SPD model, followed by some 2L-SPD practical applications. Then, we introduce the 2L-SPD variant with LIFO constraints which prohibits rearrangement of items along the vehicle trips.

2.1. Description of the basic 2L-SPD model

Let $G = (V, A)$ be a complete graph where $V = \{0, 1, .., n\}$ is the vertex set and A is the set of arcs (i, j) connecting every pair of distinct vertices. Each arc $(i, j) \in A$ is associated with a cost c_{ij} equal to the distance that must be traveled for moving from vertex i to vertex j. Vertex 0 represents the warehouse which acts as the base station of k homogeneous vehicles. Each vehicle has a maximum carrying weight equal to Q and a loading surface of length and width equal to L and W, respectively. Vertex set $N = V \setminus \{0\}$ corresponds to the customer set. With each customer $i \in N$, there are two associated item sets, namely D_i and P_i. Set D_i corresponds to the items that must be delivered from the warehouse to the customer, whereas P_i contains the items that must be picked-up from customer i and transported to the warehouse. All transported items are considered non-stackable. The total weight of item sets D_i and P_i are equal to d_i and p_i, respectively. The length and width dimensions of an item $j \in D_i \cup P_i$, $(\forall\, i \in N)$ are denoted by l_j and w_j, respectively.

The 2L-SPD model calls for the production of the route set that minimizes the total travel distance. The routes are subject to the following constraints:

- *a.* The size of the produced route set does not exceed k (at most one route assigned to each vehicle).
- *b.* Each route starts from the warehouse visits customers and returns back to the warehouse.
- *c.* Each customer is visited once by exactly one route.
- *d.* The delivery and pick-up requests of each customer are fully satisfied.
- *e.* The carrying weight of each vehicle does not exceed the maximum carrying weight Q at any point of the produced routes.
- *f.* For each of the traveled arcs, there exists a feasible packing pattern for the transported items.

Constraint (*e*) corresponds to the classic one-dimensional constraint incorporated in most of the vehicle routing variants, whereas constraint (*f*) introduces the two-dimensional loading requirements of the 2L-SPD model. This constraint is aimed at developing feasible item arrangements under the following limitations:

- *f.1* All items are placed within the loading surface (no item exceeds the loading surface).
- *f.2* There is no pair of items that overlap each other.
- *f.3* All items are packed orthogonally (their length and width edges are parallel to the length and width edges of the vehicle loading surface).

At this point, we would like to distinguish between two distinct configurations of constraint *f.3*. The first configuration (*Oriented*) dictates that items must be loaded with fixed orientation, i.e. the length dimension of any item is parallel to the length dimension of the loading surface. The second configuration (*Rotations*) allows 90° rotations of items. Using a similar typology to the one of Fuellerer (2009) for the 2L-CVRP, the *Oriented* version of 2L-SPD is denoted as 2|O|SPD, whereas the *Rotations* version is referred to as 2|R|SPD.

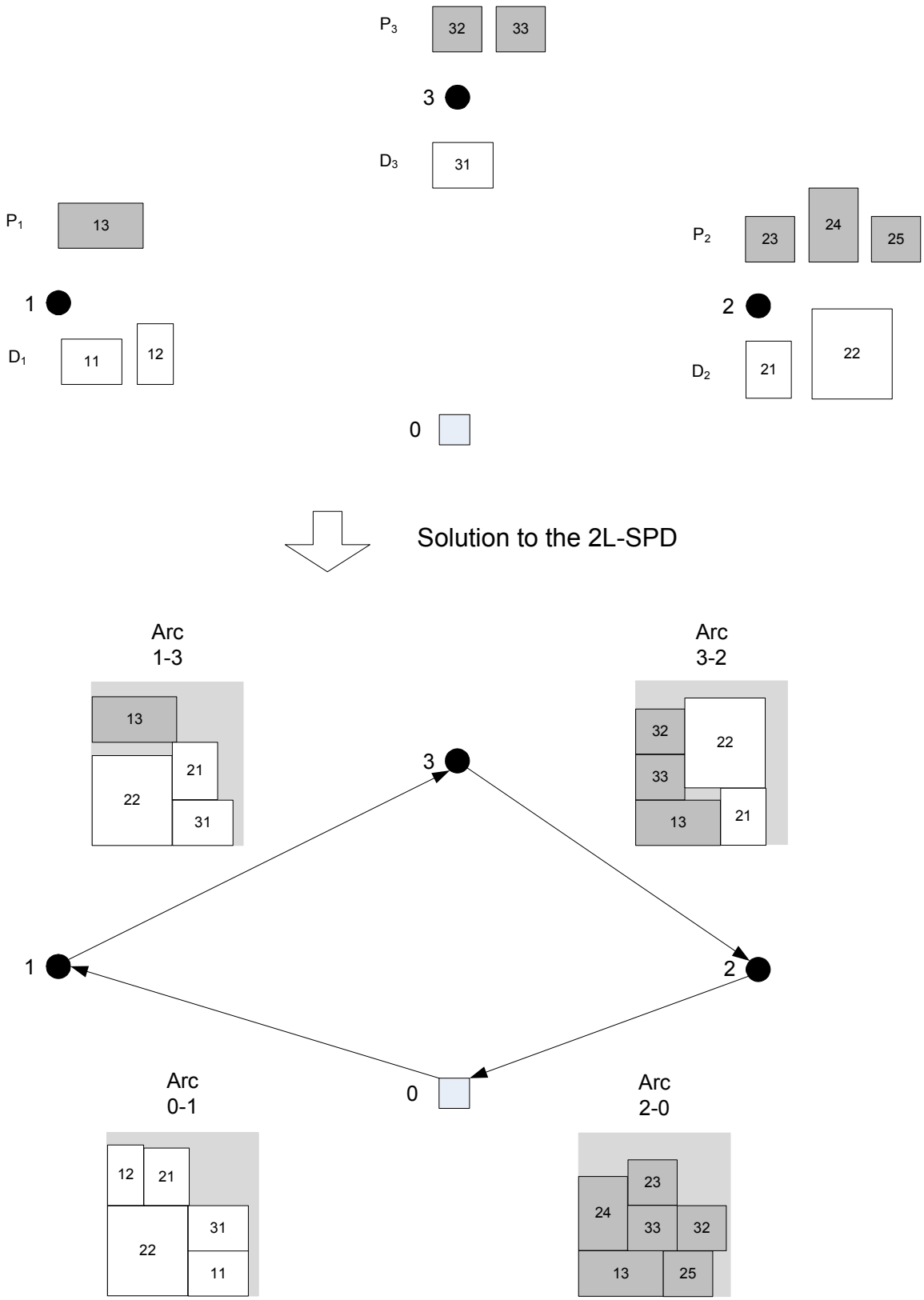

Figure 1. An example 2L-SPD solution

To better describe the model, we provide Figure 1, which illustrates an example solution for a 2L-SPD instance of one vehicle, three customers, five delivery and six pick-up items. Observe that one route has been designed for the single available vehicle. Along the route, the size of the transported item set does not monotonically decrease. Instead, some items are delivered to the customers visited and some items are picked-up to be transported back to the warehouse. Thus, feasible loading patterns must be identified for every arc traveled, to ensure that loading feasibility is guaranteed. This is the crucial difference of the 2L-SPD model compared to the 2L-CVRP one: Under the 2L-CVRP, all pick-up items sets are empty. Thus, the feasibility of the k fully loaded depot-adjacent arcs must be examined. In the case of the instance presented in Figure 1 (k = 1), the only feasibility investigation would involve arc (0, 1). On the other hand, under the 2L-SPD model which considers non-empty pick-up item sets, feasible item loadings must be determined for each solution arc. This means that in the general case of n customers and k vehicle routes, $n + k$ feasible two-dimensional packing arrangements must be determined for the transported items. For the example case (n = 3, k = 1), four loading patterns are identified, as depicted in Figure 1.

The 2L-SPD model is applicable in transportation activities, where bi-directional product flows between the warehouse and customers must be made, as dictated by reverse logistics policies. It covers cases where shipments correspond to items of various sizes which are considered non-stackable. This transportation scenario arises in the context of haul-away service offered by furniture and household appliance stores. Under this practice, the store is responsible for dispatching vehicles to deliver the purchased items to customers. In addition, the vehicles must also collect the items which customers require to dispose of. Another 2L-SPD application emerges in grocery store and supermarket networks, where goods must be replenished, while at the same time outdated products, empty pallets and roll cages must be gathered and sent to the warehouse for further processing and reuse. The aforementioned cases do not necessarily involve items of identical sizes: supermarkets may use large pallets for light products, such as paper and plastics, while bottles and food products are packed into smaller pallets, or roll cages to be easily handled when unloaded. Stacks of plastic boxes may also be used for fruits and vegetables. Another case where a mix of different pallet sizes can be used arises when networks consisting of retailers with different storage room characteristics are visited. In this context, small stores located in urban areas may prefer small and easily maneuverable pallets, whereas bigger stores located outside the city centers can use the standard sized pallets. In addition, to increase vehicle space utilization, empty stackable roll cages are folded and grouped together into shapes of various sizes in order to be sent back to the warehouse. Additional examples of 2L-SPD applications may involve cases where a customer requires either delivery or pick-up service, just as in the case of the Vehicle Routing Problem with Mixed Pick-ups and Deliveries (VRPMPD) which incorporates one-dimensional capacity constraints (Berbeglia et al. 2007). For example, the distribution of products from a production site to a set of retailers and the concurrent collection of raw materials from suppliers located within the same geographic region to replenish the production site can be modeled by 2L-SPD. Depending on the type of the delivery and pick-up materials, the transported products may correspond to box stacks or pallets of various dimensions. In the same context, office and household furniture rental businesses need to dispatch their vehicles to jointly deliver and pick-up (or even replace) pieces of furniture to and from the service locations.

2.2. Incorporating LIFO constraints into the basic 2L-SPD model

An important characteristic of the basic 2L-SPD model is that no LIFO constraint is considered. This implies that when the vehicle visits a service location, items must be repositioned inside the loading space, in order to complete the necessary loading and unloading operations. This is an analogous assumption with the one made by the Unrestricted version of the 2L-CVRP which allows items (pallets) repositioning when the vehicle visits service locations. The time required for handling such item repositioning is compensated by the simultaneous service of two transportation request types (distribution/collection) which increases the capacity utilization of vehicles and thus reduces unproductive "empty" mileage. The simultaneous delivery and pick-up service characteristic may involve drastic rearrangement of the transported items, thus service locations should offer sufficient resources (space for emptying the carrying load, forklifts or pallet jacks) to facilitate item repositioning. Finally, since radical rearrangements of the transported pallets take place, the 2L-SPD model is best suited for light and medium duty trucks typically used in city logistics operations. It is not practical, for example, to repeatedly unload and reload the 30 to 40 pallets carried by heavy duty trucks along their service trips.

To tackle the transportation scenario where item repositioning is either prohibited (due to item sensitivity) or should be avoided (fast loading and unloading operations must take place), the basic 2L-SPD model can be extended to incorporate LIFO constraints. These constraints, also referred to as *Sequence* constraints (Iori et al., 2007), ensure that the loading and unloading of every item is directly performed without being necessary to reposition any other item onboard. Under the LIFO version of the basic 2L-SPD model, in addition to the loading constraints *f.1* - *f.3*, two extra constraints are taken into account:

- *f.4* No item is positioned between any delivery item *i* and the loading door of the vehicle, when item *i* is unloaded from the vehicle (unloading without rearrangements).
- *f.5* No item is positioned between any pick-up item *i* and the loading door of the vehicle, when item *i* is loaded into the vehicle (loading without rearrangements).

Prohibiting item repositioning has the following impact on the underlying loading constraints: contrary to the basic 2L-SPD model, where a feasible loading structure must be determined for every route arc, when the LIFO constraints are taken into account, the decision maker has to determine one loading position for every transported item (delivery or pick-up). From another viewpoint, two feasible packing structures must be defined: the first one for all delivery items and the second one for all pick-up items. Obviously, these two loading structures must ensure that the LIFO requirements are satisfied. The loading structures for intermediate points in the route can be regarded as the union of these two loadings by subtracting all items which have been delivered and all items which have not yet been picked-up. We examine two configurations for the LIFO version of 2L-SPD: 2|O|SPD-L which considers fixed item orientation and 2|R|SPD-L which allows item rotations of 90°.

3. The overall 2L-SPD solution approach

The 2L-SPD model is a very challenging problem which can be regarded as the union of two NP-hard combinatorial optimization models: one for the routing aspects (Vehicle Routing Problem) (Laporte, 2009), and

one for the two-dimensional loading requirements (Two-Dimensional Bin Packing Problem) (Lodi et al., 2002). To solve the 2L-SPD in reasonable computational times, we make use of a master local search framework described in the present Section. The local search algorithm makes use of the route loading feasibility examination procedures which are thoroughly presented in Section 4.

Our master 2L-SPD approach is a two-stage method. In the first stage, a fast constructive heuristic (§3.1) is employed for building an initial 2L-SPD solution. This solution is composed by feasible 2L-SPD routes, however it may be partial or complete, in the sense that it may not serve all customer requests. This initial solution is fed to the algorithm's second stage which constitutes the core of the proposed optimization approach (§3.2).

3.1. Constructive methodology for building an initial 2L-SPD solution

A set of *k* empty routes is initialized and a randomly chosen radius originating from the warehouse is defined. Customers are sorted in increasing order of the angle formed by the random radius and the customer locations. Then, customers are iteratively selected to be inserted to the routes available. At each iteration, we examine all possible insertion positions. The customer is inserted into the solution point which is both feasible and minimizes the additional routing cost. Note that the loading feasibility of tentative routes is determined with the use of the route loading procedure presented in Section 4, which in turn calls the packing heuristic of Section 5. These calls to the packing heuristic have been designed to be as fast as possible: only one attempt (light packing mode) is applied for building a feasible packing arrangement (this will be clarified in Sections 4 and 5, where the loading feasibility examination is described). If for any customer, no feasible insertion position is identified, this customer remains unserved. The iterative procedure terminates when trial insertions have been examined for all customers.

3.2. Local search algorithm for the 2L-SPD

The proposed solution approach is based on our previous work on the 2L-CVRP (Zachariadis et al., 2013), extended to efficiently deal with the increased number of loading sub-problems that must be solved in order to decide on the feasibility of a vehicle route. It employs a blend of three local search operators for moving between solutions. To diversify the search, a simple-structured scheme based on the aspiration criteria of tabu-search is employed.

3.2.1. Local search operators

A blend of three local search operators is applied, namely the 1-0 exchange, 1-1- exchange and 2-opt.

1-0 Exchange (Customer Relocation): A move defined by the 1-0 exchange operator removes a customer from its current service position and reinserts a customer into any other solution position. In the general case, a 1-0 exchange move replaces three solutions arcs of the candidate solution. This operator is employed both within a route and between any route pair.

1-1 Exchange (Customer Swap): A move defined by the 1-1 exchange operator swaps the service positions of any pair of customers of the candidate solution. In the general case, four solution arcs are replaced. Moves defined by the 1-1 operator are employed both within a single route, as well as between any route pair.

2-opt: A move defined by the 2-opt operator replaces any pair of arcs that is included in the candidate solution. A different solution modification mechanism is followed according to whether the move is applied within a single route or between a route pair: If the 2-opt move is performed within a route, two route customers are selected and the path between these selected customers is reversed. If the 2-opt move is performed between a route pair, both routes are divided into an initial and a terminating segment. The starting segment of the first route is connected to the terminating segment of the second route and vice versa. Any combination of route division points is considered. This inter-route operator is commonly referred to as 2-opt*.

3.2.2. The SMD representation of the local search moves

As with most local search implementations, the required computational burden mainly depends on the evaluation of the neighborhood structures defined by the employed local search operators. To accelerate this decisive aspect of the proposed approach, we use the concept of Static Move Descriptors (SMD). The basic principle of the SMD strategy is that every tentative local search move defined by the employed operators is statically encoded into an SMD instance. This SMD instance encodes the structural modification of the corresponding local search move. In addition, it includes the objective function change that this move would cause, if applied to the candidate solution. Each time a move is applied to a candidate solution, a limited subset of the solution characteristics is modified. Thus, only the subset of the SMD instances which are related to this modified solution part needs to be re-evaluated according to the modified solution state. The remaining SMD instances stay valid, so their cost recalculation is redundant. Using the SMD concept, redundant local search cost recalculations are eliminated. For more details, the interested reader is referred to the article of Zachariadis and Kiranoudis (2010) where the SMD strategy was originally introduced.

Except for the objective change information, the SMD instances have been designed to include the loading feasibility status of the encoded local search moves. This allows the algorithm to eliminate any redundant calls to the time consuming loading feasibility procedures of Sections 4 and 5. This aspect will be thoroughly described in Section 4 (Level 1), where the loading feasibility of local search moves is discussed.

3.2.3. The adopted diversification component

The proposed local search exhaustively explores the solution neighborhoods defined by the local search operators and implements the move which incurs the minimal objective function change. This move selection criterion entraps the algorithm in the first local optimum (in respect to the local search operators) encountered. To avoid this situation and induce additional diversification in the search process, we make use of a mechanism which effectively filters out cycling causing local search moves. The proposed scheme is inspired by the aspiration criteria used in tabu-search implementations. It associates a cost tag with each problem arc. Let p_i denote the tag of arc $i \in A$. In addition, let E_m and C_m denote the solution arcs to be eliminated and created, respectively, if local search move m is applied to the candidate solution. When move m is applied to a candidate solution S of cost $z(S)$, the cost tag of each of the eliminated arcs is set equal to the objective function of solution S ($p_i = z(S), \forall\, i \in E_m$). During later stages of the search, a move m is allowed to be applied to a solution S for generating S', if and only if the objective of the tentative solution S' improves the cost tags of all the arcs to be created ($p_i > z(S'), \forall\, i \in C_m$). To control the diversification effect caused, the cost tags of the solution arcs are initialized to $+\infty$ every φ main algorithmic iterations ($p_i = +\infty, \forall\, i \in A$). After preliminary experiments, we have

set $\varphi = \lceil n/2 \rceil$ for which a satisfactory performance was observed. The employed diversification component can be seen as a variation of the Attribute Based Hill Climber (ABHC) (Whitley & Smith, 2004) with the following two main differences: *a*) The threshold tag for each arc is the objective of the last solution using this arc, whereas under the ABHC, the arc threshold tag is set to the score of the best solution using this arc; *b*) a move is considered admissible, if all arc thresholds are satisfied, while ABHC allows a move to be performed, if only one arc threshold is met.

3.3. The proposed local search algorithm

The core of the optimization procedure is fed with the initial 2L-SPD solution S_0 generated in §3.1, and the set of non-served customers U. Obviously, if the initial solution S_0 is complete, set $U = \emptyset$. The candidate solution S is set to be equal to S_0. Then, an iterative procedure is applied for performing structural modifications on the candidate solution. These modifications must satisfy the loading constraints of 2L-SPD. If the set of non-routed customers U is non-empty, each iteration tries to insert any not served customer into the candidate solution.

More specifically, each algorithmic iteration involves the sequential execution of the following steps:

1. The solution neighborhoods defined by the employed operators are exhaustively explored. For each operator, the move that respects the diversification scheme of §3.2.3, leads to the generation of feasible 2L-SPD routes and minimizes the objective change is identified. This step is performed by examining the SMD instances which encode the local search moves.
2. If any of the three moves improves the objective function of the candidate solution, the highest-quality of these moves is selected to be applied. If none of these moves is objective improving, one of them is selected randomly to be applied. Let m denote the selected local search move and S' the solution to be obtained if m is applied.
3. The cost tags of the eliminated solution arcs are appropriately set ($p_i = z(S), \forall\, i \in E_m$).
4. Move m is applied to the candidate solution. Thus, the candidate solution is set equal to S'.
5. The costs of the affected SMD instances are recalculated according to the updated solution S.
6. Any unserved customer contained in U is attempted to be feasibly inserted into any point of S.

Recall that each time φ iterations are executed, the arc cost tags are re-initialized. The overall procedure terminates after the completion of 50,000 iterations by returning the best complete solution generated through the search process.

At this point, we would like to mention that Step 1 is the most demanding regarding the required computational effort: The costs of tentative moves are efficiently retrieved from the corresponding SMD instances and the necessary checks regarding the adopted diversification scheme are straightforwardly performed in constant time. However, the loading feasibility investigation of tentative moves is a very complex task which must be appropriately designed, in order to be as fast as possible. A poor algorithmic design for this task practically makes the algorithm incapable of effectively tackling even small-scale 2L-SPD instances. The proposed procedure for examining the loading constraints for a local search move is analyzed in Section 4, which also presents the memory structures employed for recording obtained loading feasibility through the search process.

4. Loading feasibility examination of local search moves

The evaluation of the loading feasibility of local search moves is the most time consuming task repeatedly executed through the 2L-SPD solution approach (Step 1 of §3.3). In the following, we provide an analytic description of the three employed examination levels for obtaining the loading feasibility status of a local search move. In addition, for each of these levels, the proposed memory components used for recording loading feasibility information and accelerating the search are presented.

We start by introducing the employed data structures and relevant notation:

- m: the local search move whose loading feasibility is investigated.
- $z(m)$: the objective function change that this move incurs if applied to the candidate solution.
- SMD_m: the SMD instance encoding local search move m.
- R_m: the set of routes to be affected by move m. Note that $|R_m| = 1$, if m is an intra-route move, whereas $|R_m| = 2$ if m is an inter-route move.
- *RH*: The hashtable used to record "strong" route loading feasibility examinations.
- *RHL*: The hashtable used to record "light" route loading feasibility examinations.
- *AH*: The hashtable used to record "strong" item packing feasibility examinations.
- *AHL*: The hashtable used to record "light" item packing feasibility examinations.

We note that the terms "*strong*" and "*light*" are used to characterize the mode of the employed feasibility examination. The strong mode increases the probability of declaring a local search move feasible. It is used for objective improving local search moves. On the other hand, the light mode is considerably faster and it is used for non-cost improving moves. The feature that distinguishes these two modes is related to the maximal packing attempts (parameter μ) used for the calls to the packing heuristic of Section 5. This issue is further discussed when the third feasibility examination level is provided.

Level 1. Loading feasibility examination of local search moves

The first level of evaluating the feasibility of a local search move was briefly discussed in §3.2.2 where the SMD representation is presented. Each SMD instance contains two pieces of information: a) a binary value representing if the encoded move was found to be feasible or not the last time it was checked, and b) an integer value equal to the master algorithmic iterator, when this last feasibility check was performed (iterative procedure of §3.3). Thus, when a move m has to be examined regarding the loading constraints, the following steps are applied: we examine if the routes involved in R_m have been modified since the last time that the loading feasibility of m was examined. Note that this is straightforwardly implemented by associating a counter with each route corresponding to the algorithmic iterator, when this route was last modified. If the routes of R_m have remained unmodified, then the feasibility status of the examined move is directly retrieved from the binary flag contained in SMD_m. Otherwise, the steps of Level 2 are performed.

Level 2. Loading feasibility examination of complete routes

The second feasibility examination level is associated with the loading feasibility check of complete routes. Before we provide the tasks performed for examining the route loading feasibility, we present hashtable structures *RH* and *RHL*.

Each entry of the *RH* and *RHL* hashtables corresponds to the loading feasibility status of a route. The key of each entry corresponds to the string representation of a given route. To prepare this representation, the IDs of customers visited by these routes are concatenated and separated by a standard character (i.e. '*'). For example, the string representation of the route depicted in Figure 1 is "1*3*2". It is important to note that the sequence of customers remains as is. No sorting takes place, as the loading feasibility is determined for a given customer permutation (not combination). This is because each customer permutation defines a unique set of arcs for which the packing feasibility must be examined. The value of each entry corresponds to a binary flag indicating if this route has been found to be feasible or not. *RH* is responsible for storing the feasibility obtained by employing "strong" loading checks, whereas *RHL* is used to store feasibility information obtained by "light" loading checks.

Although the *RH* and *RHL* hashtables are designed to record feasibility information obtained by different modes of the loading examination procedures, their contents must conform to the following rules:

1. If a route is found to be feasible by any examination mode (strong or light), the corresponding information is kept in both *RH* and *RHL*
2. If a route is found to be infeasible by the strong examination mode, the corresponding information is kept in both *RH* and *RHL*.

We point out that if a route is declared infeasible by a light examination, the corresponding entry is recorded in *RHL*. To be consistent with rule (1), if this route is re-examined via the strong examination mode and is found to be feasible, the corresponding entry is pushed in *RH*. The relevant entry stored in *RHL* is updated by changing the binary flag from false to true.

The second level for examining the feasibility of move m starts off by generating the string representation of the routes contained in R_m. Then, two cases are distinguished according to $z(m)$:

If $z(m) < 0$ (objective improving move), for each R_m route, we check if the corresponding entry is contained in *RH*. If such entries exist, the loading status of the R_m routes is retrieved directly from the corresponding hashtable values. If any route is found to be infeasible, move m is declared infeasible, whereas if all R_m routes are found to be feasible, move m is declared feasible. If for any R_m route, no relevant entry is contained in *RH*, then for this route, we have to evaluate the loading feasibility by moving to Level 3 and using the strong mode of loading examination.

If $z(m) \geq 0$ (objective augmenting move), for each R_m route, we retrieve the corresponding feasibility values contained in *RHL*, exactly as described for the *RH* hashtable. If for any R_m route, no relevant entry is contained in *RHL*, then for this route, we have to evaluate the loading feasibility by moving to Level 3 and using the light loading examination mode.

Level 3. Loading feasibility examination of the arcs traveled by a route

The third loading feasibility level is related to the feasibility of the individual arcs contained in a route. Before we provide the tasks performed for examining the loading feasibility of route arcs, we present the hashtable structures *AH* and *AHL*.

Each entry of the *AH* and *AHL* hashtables corresponds to the loading feasibility status of a given set of items. The key of each entry is a string which indirectly defines an item set. Under the 2L-SPD model, transported item sets are the union of some customers' delivery items and some customers' pick-up items. Let's take a closer look at this: whenever a vehicle traverses an arc (*i*, *j*), it carries the pick-up demands of customer *i* and all of its predecessors, and the delivery items of customer *j* and all of its successors. Thus, under the 2L-SPD model, any transported set of items may be fully described by two customer sets: one associated with the pick-up and one associated with the delivery service. This is the basic idea for preparing the string representation of a given item set: Both sets are individually sorted according the customer IDs. Then for each set a string representation is straightforwardly prepared by using a standard separator character (i.e. '*'). These two strings are then concatenated (the delivery string is placed first) and separated by another character (i.e. '-'). The resulting string fully describes the item set carried along an arc. Under the aforementioned rationale and for the example case of Figure 1, the string representations of the four item sets are as follows: Arc (0,1): "1*2*3-", Arc (1,3): "2*1-3", Arc (3,2): "1-2*3" and Arc (2,0): "-1*2*3". Regarding the sorting according to the customer IDs, this is applied because we are interested in the item *combinations*, so that the relative positioning of customers within the strings is irrelevant. The value of each *AH* and *AHL* entry corresponds to a binary flag indicating if the corresponding item set has been found to be feasible or not. *AH* is responsible for storing the feasibility obtained by employing the "strong" packing mode, while *AHL* is used to store feasibility information obtained by "light" packing checks.

Despite the fact that *AH* and *AHL* are designed to record loading feasibility obtained by different modes of the packing heuristic, their contents must conform to the following rules:

1. If for an item set, any packing heuristic mode (strong or light) generates a feasible loading pattern, the corresponding information is kept in both *AH* and *AHL*.
2. If for an item set, the strong mode of the packing heuristic cannot generate a feasible packing arrangement, the corresponding information is kept in both *AH* and *AHL*.

At this point, we note that if an item set is found infeasible by the light mode of the packing heuristic, the corresponding entry is recorded in *AHL*. If the same item set is re-examined by the strong examination mode and is found feasible, the corresponding entry is inserted in *AH*, whereas the corresponding entry in *AHL* is updated to indicate that the item set is feasible (entry value is set to true), to be consistent with rule (1).

Let *r* be the route that must be evaluated in terms of the 2L-SPD loading feasibility. In addition, let A_r be the set of arcs traveled by route *r*. For each arc $i \in A_r$, let S_i denote the set of transported items. In addition let $a_i = \sum_{j \in S_i}(w_j \cdot l_j)$ denote the total area of the item set S_i.

The procedure for evaluating the feasibility of a route *r* begins by sorting A_r in decreasing order of the total area of the corresponding item sets (a_i). In addition, an empty set *H* is initialized. Then, arcs are selected one by one

and the following sequence of tasks is performed: Let *i* denote the selected arc. The string representation of the item set S_i is generated. If the "weak" examination mode is performed, the procedure checks if the packing feasibility of S_i is recorded in *AHL*. If this is the case, the packing feasibility is directly retrieved from *AHL*. If item set is found to be infeasible, the whole Level 3 procedure terminates by declaring route *r* infeasible. If no S_i entry exists in *AHL*, arc *i* is pushed in the set *H*. Under the "strong" examination mode, the aforementioned steps are followed, but the loading feasibility information is retrieved from hashtable *AH*.

If after this first arc pass, the *H* set is non-empty, this means that there are item sets whose loading feasibility could not been directly retrieved from the hashtable structures. For these arcs, the feasibility must be determined by the packing heuristic of Section 5. To do so, the *H* arcs are picked one-by-one in the order that they were pushed in *H*. Let *i* denote the selected arc. If the "weak" mode is used, then the heuristic of Section 5 is employed for the item set S_i using just a single packing attempt (μ = 1). On the contrary, if the "strong" mode is used, the heuristic is applied to the S_i items by performing up to 1,000 packing attempts (μ = 1,000). Note that the aspect of the maximum number of packing attempts is clarified in Section 5, where the packing heuristic is described. Obviously, the obtained packing information is appropriately recorded in hashtables *AHL* and *AH*. In addition, if for any arc, no feasible packing arrangement is found, route *r* is declared infeasible and the loading examination is terminated. Otherwise, if for every *H* arc, feasible packing arrangements for the corresponding item sets are identified, route *r* is considered feasible.

5. Examining the loading feasibility of an item set

The core of the 2L-SPD loading feasibility investigation consists of constructing feasible two dimensional orthogonal packings for given item sets. As already stated, a feasible packing structure must be identified for every solution arc, thus the task of identifying feasible arrangements for given item sets is repeatedly executed within the overall 2L-SPD solution approach. For this reason, the proposed two-dimensional packing procedure (hereafter called *packing heuristic*) was mainly designed for computational speed. The basic characteristic of our two-dimensional packing heuristic is that a series of attempts for feasibly packing every item is performed. Each attempt successively inserts items in the vehicle loading space. The proposed two-dimensional packing heuristic is based on the procedure employed for the 2L-CVRP (Zachariadis et al., 2013) extended to consider additional loading positions, as will be thoroughly described.

5.1. Availability of loading positions

Each item can be inserted into a set of candidate loading positions. To designate these positions, we use a Cartesian coordinate system (*w*, *l*) defined by the edges of the loading surfaces. Let the origin of the axes (0, 0) corresponds to the backmost and leftmost position of the loading surface, whereas the vehicle loading door is defined by the linear segment originating at (0, L) and terminating at (W, L).

The loading position set is updated according to the item insertions that take place, similarly to the extreme point procedure of Crainic et al. (2008). Specifically, when a packing attempt begins, the vehicle space is empty and the only loading position available is at position (0, 0). Each time an item is loaded into the surface, the corresponding insertion position is removed from the available position set. It is substituted by two loading

positions which are generated in the left-front and right-back corners of the inserted item. Some additional positions are generated by employing the four mechanisms depicted in Figure 2. The first three mechanisms were incorporated in our 2L-CVRP approach (Zachariadis et al., 2013), whereas the fourth mechanism extends this former method by offering additional placement positions. Briefly, the first mechanism applies the following rationale: when an item *E* is placed in a position, its right side is projected and a new position is created at the intersection of this projection with the nearest item placed behind *E* (position p_3). Similarly, item's *E* front edge is projected and a new position (p_1) is created at the intersection of this projection with the nearest item placed on the left side of *E*. Regarding the second mechanism, when an item *C* is inserted, we look for already placed items lying on the front side of *C* whose right side projections intercept the front side of *C* (item *E*). New placement positions are created on the intersection of these projections with the front side of *C*. Similarly, under the third mechanism, we look for already placed items lying on the right side of the inserted item *D* whose front side projections intercept the right side of *D* (item *E*). New placement positions are created on the intersection of these projections with the right side of *E*. Regarding the fourth mechanism, it is based on the envelope approach (2D-CORNERS) introduced by Martello et al. (2000). It is used to define additional loading positions where the envelope of inserted items changes from vertical to horizontal. For the case depicted in Figure 2, the new position is located at the intersection of the right edge projection of item *D* and the front edge projection of item *B*. This position would be missed by the first three mechanisms. Loading positions generated according to the envelope-based mechanism can be of major importance, especially when few items of significant size (relatively to the *L* and *W* dimensions) must be packed into the loading surface. Note that the four mechanisms of creating loading positions may lead to duplicate insertions positions. These duplicate positions are avoided by appropriate checks. In addition, loading positions contained in the areas occupied by inserted items are removed from the set of available insertion positions, as they cannot accommodate subsequent items.

5.2. Memory components for diversifying the packing arrangements

As previously mentioned, the methodology for determining the loading feasibility of an item set performs a series of attempts to successively pack all items into the loading space. In general, these attempts should be aimed at building diverse packing arrangements to maximize the overall probability of obtaining a feasible complete loading pattern. To systematically promote the development of diverse patterns, we employ a memory mechanism for recording the frequency of encountered partial loadings. This memory component is implemented as a hashtable. The key of each hashtable entry encodes a specific packing arrangement, while the entry value gives the number of times that this arrangement has been developed through the packing heuristic. In terms of the hashtable keys, a straightforward procedure for mapping packing structures to strings has been followed. It extends the one presented in Zachariadis et al. (2013) by considering a binary flag to indicate if an item is rotated. Each string uses two standard separator characters: character '*' separates individual item packing information, while character '-' separates two distinct items. The general format of a string that encodes a packing of q items is (ID_1*pw_1*pl_1*r_1- … - ID_q*pw_q*pl_q*r_q), where ID_i ($1 \le i \le q$) denotes the unique identifier of the i-th inserted object, pw_i and pl_i denote its placement coordinates (back left corner) and binary r_i indicates if this item is rotated. The mapping between loadings and their string representation is depicted in Figure 3. When character 'R' is reported next to the item ID, this implies that the item is rotated. Note that Figure 3 illustrates the first packing attempt. This is why all hashtable values are set to 1. In subsequent packing attempts, if any of these partial packing patterns is re-encountered, the corresponding value will be augmented accordingly.

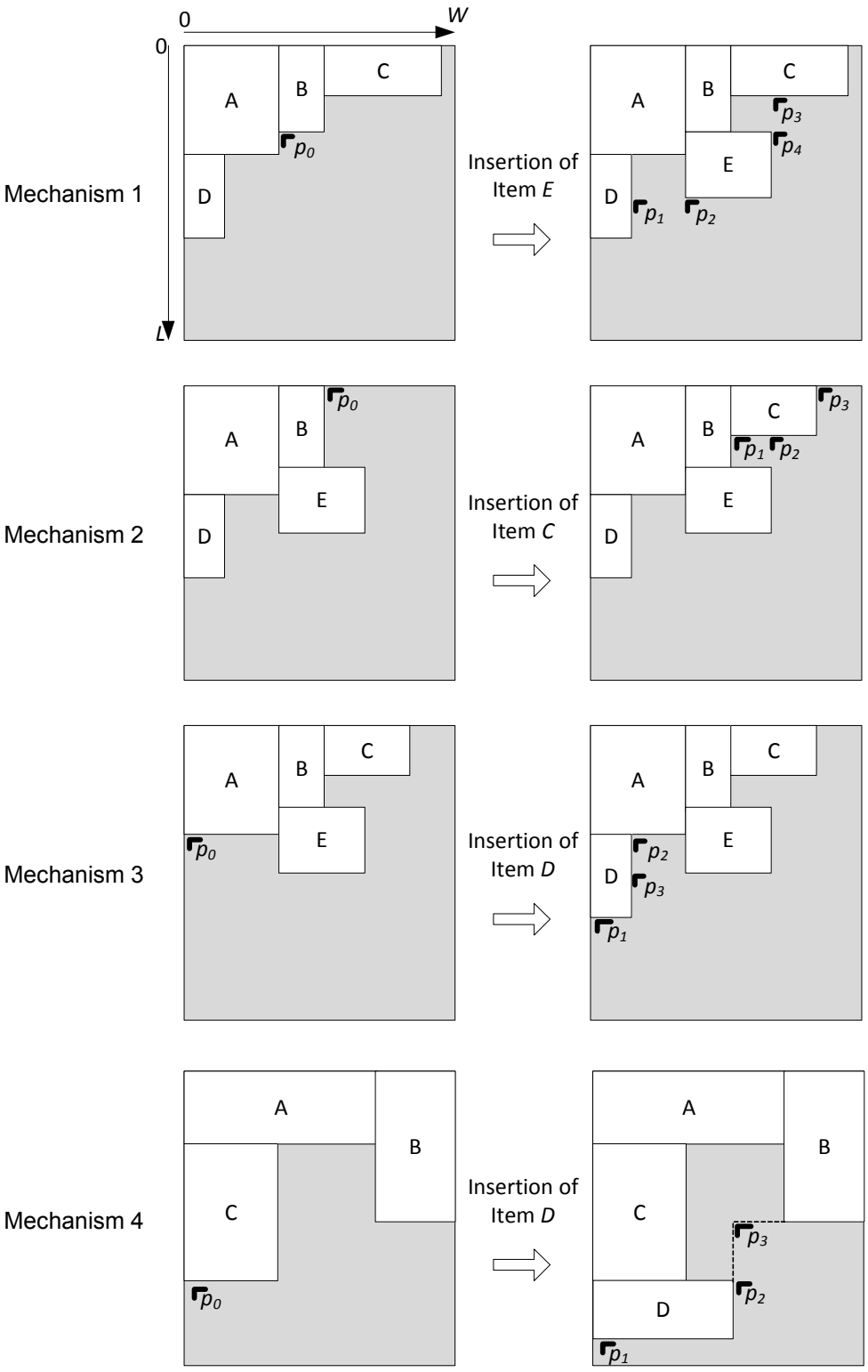

Figure 2. The mechanisms employed for updating the available loading positions

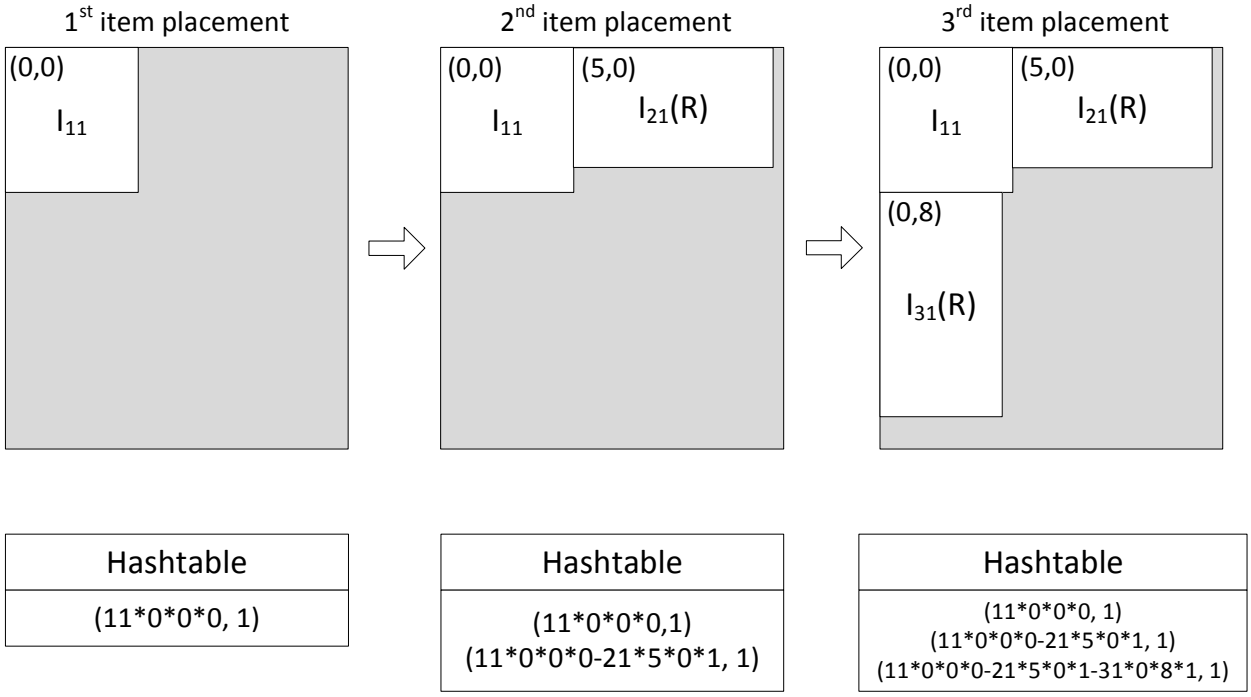

Figure 3. Hashtable Structures for recording the frequency of partial loading structures

5.3. The procedure for examining an item set loading feasibility

When the loading feasibility of an item set *T* has to be evaluated, the adopted packing heuristic performs a series of attempts for feasibly loading all items into the loading surface (Zachariadis et al., 2013). Each attempt is initiated by setting the set of available loading positions to {(0, 0)}. Then, an iterative procedure begins for successively inserting items into the loading surface. Each of these iterations consists of the four following steps:

- *Step 1 - Identification of the optimal item insertion*
- *Step 2 - Insertion of the item in the selected position*
- *Step 3 - Update the set of available loading positions*
- *Step 4 - Update of the memory components used for recording the frequency of partial loadings*

In the following, we provide a brief description of each of these steps. This description makes use of the following notation: *U* denotes the non-loaded items, whereas *P* denotes the set of available loading positions. Set *R* corresponds to the set of rotation indices {0, 1}.

Step 1 - Identification of the optimal item insertion

Regarding the first step of identifying the optimal item-loading position-rotation index placement, it is responsible for deciding which item and where this item is going to be inserted. In addition it determines if the item is going to be inserted rotated or not. For each of the yet non-loaded items $i \in U$, rotation indices $j \in R$ and *feasible* (satisfying all loading constraints) insertion positions $k \in P$, the following utility function is calculated:

$$u(i, j, k) = TP(i, j, k) - \lambda \cdot M \cdot t(i, j, k) \quad (1)$$

where $t(i, j, k)$ denotes the number of times that the partial loading to be generated, if item i is placed according to the rotation index j in position k, has been encountered through all attempts of the packing heuristic. This value is retrieved from the hashtable structure presented in §5.2. In addition, λ is a binary parameter which controls the diversification of the packing arrangements, as will be discussed in the following and M is a large positive value. $TP(i, j, k)$ denotes the total touching perimeter of item i with either already placed items or surface boundaries, if it is inserted in position k according to the rotation index j.

The insertion triplet item i – rotation index j – position k which maximizes the utility function (1) is selected to be applied. The proposed utility function extends the one presented in our previous study (Zachariadis et al., 2013) by allowing item rotations. It is made up by two terms: The first term is directly associated with the Maximum Perimeter heuristic (Lodi et al., 1999). The second term is used to diversify the obtained loading arrangements at each loading attempt. Binary parameter λ is used to control this diversification effect. At each iteration, if $\lambda = 0$ (probability d), the item-position pair is decided solely on the touching perimeter criterion. On the other hand, if $\lambda = 1$ (probability $1 - d$), the item-position-rotation index triplet is decided according to the frequency of partial item arrangements. More specifically, the least frequent packing arrangements are promoted. Ties are broken with the use of the touching perimeter metric. Parameter d was fixed at 0.25, as preliminary experiments indicated that the packing heuristic has better chances of generating feasible loadings when diverse packing arrangements are explored.

Step 2 - Insertion of the item in the selected position

The selected item is removed from the set of non-loaded items U and inserted into the selected insertion position according to the selected rotation index.

Step 3 - Managing the set of available loading positions

Depending on the insertion implemented in the previous step, the set of available loading positions is modified according to the mechanisms presented in §5.1.

Step 4 - Update of the memory components used for recording the frequency of partial loadings

The hashtable structure presented in §5.2 is updated according to the partial loading pattern obtained after the item has been appropriately placed into the selected loading position.

If during Step 1, for any item no feasible position is identified, this implies that the current packing attempt cannot produce a feasible loading arrangement. Thus, the present packing attempt is aborted, the loading surface is emptied and the next packing attempt is employed from the beginning. Of course, the hashtable structure remains unmodified, as all packing attempts are interconnected by the information stored in the hashtable.

The overall packing heuristic is terminated in two distinct cases: If a complete feasible loading structure is identified, the examined item set T is declared feasible. Otherwise, if a maximum number of μ unproductive

loading attempts are performed, the heuristic terminates by declaring the examined item set *T* infeasible. As already stated in Section 4, the light mode of the packing heuristic employs just a single packing attempt (μ = 1), while under the strong heuristic mode, up to 1,000 packing attempts are performed (μ = 1,000).

6. Methodological Modifications for the 2L-SPD with LIFO constraints

To tackle the 2L-SPD model with LIFO constraints, we apply the master algorithmic framework presented in Section 3. Obviously, the altered loading constraints require some relevant modifications on the procedures for examining the feasibility of tentative moves and packing constraints for a given item set, which are presented in Sections 4 and 5, respectively. The basic rationale of the heuristic procedure designed to deal with the LIFO version of the 2L-SPD is that the vehicle space is divided in two separate corridors (compartments): one for the delivery and one for the pick-up items. Within each of these compartments feasible loading structures must be determined taking into account the LIFO requirements of the problem.

6.1. Loading feasibility examination of local search moves for the LIFO version

The first two levels of feasibility investigation are performed exactly as described for the basic 2L-SPD version. The third level is completely modified as a result of the different loading constraints. More specifically, for a given route *rt*, we need to determine one feasible loading arrangement for the delivery items and one for the pick-up items of this route. Let C_{rt} denote the set of customers contained in route *rt*. In addition, let $D_{rt} = \bigcup_{i \in C_{rt}} D_i$ and $P_{rt} = \bigcup_{i \in C_{rt}} P_i$ denote the delivery, and pick-up items carried along this route respectively. In addition, let D''_{rt} be the subset of D_{rt} items that are delivered before the first pick-up item is loaded onto the vehicle. Then, $D'_{rt} = D_{rt} \setminus D''_{rt}$ corresponds to the set of delivery items which will be onboard together with some pick-up items of the route. Analogously, let P''_{rt} denote the pick-up items to be collected from the service points after the last delivery item of the route has been unloaded. $P'_{rt} = P_{rt} \setminus P''_{rt}$ is the set of pick-up items that co-travel with some of the delivery items of D'_{rt}.

As already stated, the proposed procedure for investigating the loading feasibility for the LIFO 2L-SPD version is aimed at loading the delivery and pick-up items in two distinct corridors of the vehicle loading surface. These corridors are formed by means of a separating line parallel to the *L* dimension of the loading surface. Precisely, the delivery corridor is defined as the rectangular area embraced by boundary points (0, 0), (W_D, 0), (W_D, *L*) and (0, L). Consequently the pick-up corridor corresponds to the remaining loading surface area, defined by points (W_D, 0), (*W*, 0), (*W*, *L*) and (W_D, *L*). To calculate W_D, or in other words the width of the corridor dedicated for the delivery items, we use the following steps. Firstly, *we* calculate $\lambda_D = \sum_{j \in D_{rt}} (w_j \cdot l_j) \; / \; \sum_{j \in D_{rt} \cup P_{rt}} (w_j \cdot l_j)$. In addition, we compute $w^D_{min} = max_{j \in D'_{rt}} \{w_j\}$ and $w^P_{min} = max_{j \in P'_{rt}} \{w_j\}$. These dimensions correspond to the absolute minimal width required for accommodating the delivery and pick-up items, respectively. Note that if rotations are allowed, these dimensions are obtained as $w^D_{min} = max_{j \in D'_{rt}} \{min\,(w_j, l_j)\}$ and $w^P_{min} = max_{j \in P'_{rt}} \{min\,(w_j, l_j)\}$, respectively. Four cases may arise: if $w^D_{min} + w^P_{min} > W$, the examined route is declared infeasible. If $\lambda_D W \geq w^D_{min}$ and $(1 - \lambda_D) W \geq w^P_{min}$, then $W_D \leftarrow \lambda_D W$. If $\lambda_D W < w^D_{min}$, then $W_D \leftarrow w^D_{min}$. Otherwise (if $(1 - \lambda_D) W < w^P_{min}$), $W_D \leftarrow W - w^P_{min}$. After the delivery and pick-up corridors have been defined, the method tries to feasibly load all delivery and pick-up items in the vehicle. More specifically, for

the delivery items, the proposed feasibility procedure employs the following tests: The loading feasibility of packing the D_{rt} items in the delivery corridor is examined with the use of the packing heuristic method presented in Section 5. If the D_{rt} items cannot be feasibly placed in the delivery corridor, the method tries to accommodate the D'_{rt} items in the delivery corridor and the D''_{rt} items in the pick-up corridor via the packing heuristic. If no feasible loading arrangement is identified, route *rt* is declared feasible. If all delivery items are feasibly loaded onto the vehicle, then the method precedes by examining the loading feasibility of the pick-up items. More specifically, the packing heuristic is applied to feasibly load all P_{rt} items onto the pick-up corridor. If no feasible loading structure can be obtained, the method tries to load the items of P'_{rt} onto the pick-up corridor of the loading surface and the items of P''_{rt} onto the delivery corridor. If both items sets are feasibly packed, then route *rt* is deemed feasible. Otherwise, route *rt* is considered infeasible. Note that under the LIFO version of 2L-SPD, the employed packing heuristic corresponds to the one presented in Section 5, modified for tackling the LIFO requirements. These methodological modifications are discussed in the next paragraph.

6.2. Examining the loading feasibility of an item set under the LIFO version of 2L-SPD

The packing heuristic used for examining the loading feasibility for a given item set employs the same rationale, as presented in Section 5. However, it is slightly modified, in terms of Step 1. More specifically, the heuristic is tuned to ensure that the LIFO constraints are effectively tackled. This means that when the set of delivery items are packed, the items that are unloaded first should be placed near the unloading door, whereas the pick-up items that are collected early on the route should be pushed back onto the loading space. To do so, the utility function (1) employed for selecting the box - placement position – rotation index triplet is modified as follows:

$$u(i, j, k) = \mathrm{TP}(i, j, k) - \lambda \cdot \mathrm{M}_1 \cdot t(i, j, k) + \mathrm{M}_2 \cdot v_i . \quad (2)$$

Note that (2) augments (1) by the term $\mathrm{M}_2 \cdot v_i$, where M_2 is a large positive value for which $\mathrm{M}_1 >> \mathrm{M}_2$ and v_i is the visit order of item *i*. If delivery items are packed, v_i corresponds to the position of the customer associated with item *i* in the route involved. On the contrary, if the packing heuristic is applied for pick-up items, v_i is set equal to the opposite customer position in the route. This is because, for the delivery items, items unloaded late on the route should be placed first (pushed back in the loading surface). The same applies for pick-up items collected early on the route. Obviously, a combination of placement position and rotation status is considered feasible for an item, only if the LIFO constraints *f.4* and *f.5* introduced in §2.2 are satisfied.

7. Computational Results

The proposed solution approach was tested on new 2L-SPD instances which were derived from the well-known 2L-CVRP benchmark problems. Due to the fact that this is the first time that an algorithmic solution is applied to 2L-SPD, we have also performed additional experiments on the 2L-CVRP and VRPSPD models. These experiments are aimed at building confidence on the reliability and effectiveness of our revised routing and packing components.

All experiments were executed on a computer system equipped with an Intel Xeon E5-2650 v2 (2.6 GHz) processor and 16 GB of RAM. The 2L-SPD methodology, hereafter referred to as *LS-2LSPD*, was implemented in C# and ran as a single core process. All 2L-SPD test instances and relevant analytic solutions are available at http://users.ntua.gr/ezach/.

7.1. 2L-SPD benchmark instances

The new 2L-SPD instances were derived from the 2L-CVRP test cases introduced by Gendreau et al. (2008). They involve 36 graphs consisting from 15 up to 255 customers. For each of these graphs, four classes (Classes 2-5) of item characteristics are considered. The higher the class index, the more and smaller items are involved in the benchmark instance.

To derive the new 2L-SPD instances the following rationale was used: Each item of the original 2L-CVRP instance was randomly designated as a delivery or a pick-up item. To promote the generation of challenging 2L-SPD instances, we used a 50/50 probability. This implies that comparable pick-up and delivery quantities are transported, so that the utilization of the vehicle space stays high along the vehicle routes which in turn makes more difficult to examine the 2L-SPD solution feasibility. In terms of the one-dimensional delivery and pick-up weight attribute, we used the following rule: let q_i denote the original delivery weight attribute for each customer $i \in N$ in the original instances. In addition, let $ad_i = \sum_{j \in D_i}(w_j \cdot l_j)$ and $a_i = \sum_{j \in D_i \cup P_i}(w_j \cdot l_j)$ denote the total area of the delivery items and the total area of both the pick-up and delivery items of customer $i \in N$, respectively. The original weight attribute q_i was translated to a couple of attributes $d_i = \|(ad_i/a_i) \cdot q_i\|$ and $p_i = q_i - d_i$, representing the total weight of the delivery and pick-up items of customer $i \in N$, respectively. Table A.1 provides a summary of the new 2L-SPD instances.

7.2. 2L-SPD benchmark instances for the LIFO version

An additional class of new test problems were constructed and solved for the LIFO version of the 2L-SPD model. They are derived from the 2L-SPD instances of Classes 4-5 with up to 100 customers. Some necessary changes were made on the item sets, the one-dimensional delivery and pick-up order levels, as well as the characteristics of the available vehicle fleets, to ensure that the resulting test cases are both feasible and not trivial. The complete characteristics of these instances are reported in Table A.2.

7.3. Computational Results on the 2L-CVRP model

As already mentioned, the 2L-CVRP model was solved to gain insight on the effectiveness of our revised routing and packing components. More specifically, we examined: The *Oriented* configuration (2|UO|L), which considers that items must be placed with their l- and w- dimensions parallel to the L- and W- dimensions of the loading space, the *Rotations* configuration (2|UR|L), which considers that items may be rotated by 90°.

7.3.1. Results on the 2|UO|L configuration of 2L-CVRP

Each of the 144 instances was solved 10 times. The obtained results for each instance are summarized in Table A.3. More specifically, for each instance, we report the average and best solution score over the ten runs, the average computational time for obtaining the final solutions of the ten runs and the percent gap between our best and average solution scores. The proposed method exhibits a rather robust performance: the average

percent gaps between our best and average scores over the ten runs are limited to 0.45%, 0.36%, 0.34% and 0.27% for Classes 2, 3, 4 and 5, respectively. In general, the larger the scale of the instances, the higher the percent gaps observed. More specifically, the highest gaps for Classes 2, 3, 4 and 5 were observed for instances 33 (1.48%), 29 (2.16%), 28 (1.42%) and 29 (0.92%), respectively. Concerning the average CPU time required for obtaining the final solution of each of the ten runs, they averaged from 316.7 sec for Class 2 up to 409.8 for Class 5. This is because the higher the Class index, the greater the number of items involved in the test problems and thus the more computational time is spent for running the packing heuristic procedure. The computational times required by the proposed solution approach are deemed acceptable, jointly taking into account the complexity of the examined model, the large scale of the test instances and the high-quality solutions produced. Table A.4 provides the best known solution score (BKS) and the percent gap between our best and the BKS value. We note that the BKS scores are taken by the works of Duhamel et al. (2011), Leung et al. (2011), Zachariadis et al. (2013), Dominguez (2014) and Wei et al. (2015). Our method produced solutions of fine quality. In total, 12 new best solutions were generated (four for Class 2, six for Class 3, two for Class 4). In addition, our method matched the best solution scores for 56 test cases (16 for Class 2, 14 for Class 3, 13 for Class 4 and 13 for Class 5). Note that some minor reported discrepancies ($\pm$0.01) on the reported solution scores may be caused by different rounding schemes. On average, our solution scores are just 0.21% higher than the BKS scores (0.08% for Class 2, 0.12% for Class 3, 0.30% for Class 4 and 0.36% for Class 5).

Table 1. Comparison of the best performing algorithms for the 2|UO|L version of 2L-CVRP

	GRASP		PRMP		VNS		LS-2LSPD		AVG		BST		
Instance	**bst**	**avg**	**bst**	**avg**	**bst**	**avg**	**bst**	**avg**	g_{PRMP}	g_{VNS}	g_{GRASP}	g_{PRMP}	g_{VNS}
1	282.66	-	281.23	281.23	281.23	281.23	281.23	281.23	0.00	0.00	-0.50	0.00	0.00
2	339.26	-	339.26	339.35	339.26	339.26	339.26	339.26	-0.03	0.00	0.00	0.00	0.00
3	376.32	-	376.32	376.32	376.32	376.32	376.32	376.32	0.00	0.00	0.00	0.00	0.00
4	435.00	-	435.00	435.12	435.01	435.01	435.00	435.00	-0.03	0.00	0.00	0.00	0.00
5	379.03	-	379.03	379.03	379.03	379.03	379.03	379.03	0.00	0.00	0.00	0.00	0.00
6	497.05	-	497.05	497.13	497.05	497.05	497.05	497.05	-0.02	0.00	0.00	0.00	0.00
7	691.11	-	690.68	690.68	690.68	690.68	690.68	690.68	0.00	0.00	-0.06	0.00	0.00
8	678.84	-	678.84	679.74	678.84	679.26	678.84	678.84	-0.13	-0.06	0.00	0.00	0.00
9	612.01	-	612.01	612.84	612.01	612.01	612.01	612.01	-0.14	0.00	0.00	0.00	0.00
10	675.79	-	676.75	676.75	674.92	675.38	676.73	676.73	0.00	0.20	0.14	0.00	0.27
11	705.95	-	703.22	703.22	702.47	704.94	703.22	705.46	0.32	0.07	-0.39	0.00	0.11
12	611.26	-	611.26	611.26	611.20	611.21	611.26	611.26	0.00	0.01	0.00	0.00	0.01
13	2490.63	-	2491.18	2491.18	2484.16	2491.31	2491.18	2491.23	0.00	0.00	0.02	0.00	0.28
14	984.42	-	975.88	979.29	975.07	976.33	974.76	975.54	-0.38	-0.08	-0.98	-0.11	-0.03
15	1144.69	-	1132.91	1134.95	1128.60	1131.02	1130.36	1133.93	-0.09	0.26	-1.25	-0.22	0.16
16	699.80	-	699.80	699.80	699.80	699.80	699.80	699.80	0.00	0.00	0.00	0.00	0.00
17	864.06	-	864.06	864.62	864.06	864.06	864.06	864.21	-0.05	0.02	0.00	0.00	0.00
18	1029.72	-	1031.95	1031.95	1027.98	1029.32	1030.98	1033.31	0.13	0.39	0.12	-0.09	0.29
19	739.19	-	741.79	743.66	737.74	741.03	740.66	741.46	-0.30	0.06	0.20	-0.15	0.40
20	522.69	-	515.44	517.53	515.92	517.02	512.84	514.75	-0.54	-0.44	-1.88	-0.50	-0.60
21	994.58	-	992.78	998.75	991.63	993.74	992.33	997.55	-0.12	0.38	-0.23	-0.05	0.07
22	1021.45	-	1023.02	1027.92	1019.03	1021.01	1018.08	1022.44	-0.53	0.14	-0.33	-0.48	-0.09
23	1038.16	-	1032.36	1036.62	1030.40	1031.99	1031.44	1034.86	-0.17	0.28	-0.65	-0.09	0.10
24	1107.94	-	1104.64	1109.00	1102.53	1103.23	1103.10	1107.81	-0.11	0.41	-0.44	-0.14	0.05
25	1345.08	-	1341.26	1347.62	1333.76	1337.18	1334.33	1341.15	-0.48	0.30	-0.80	-0.52	0.04
26	1317.41	-	1311.79	1320.11	1306.60	1309.85	1312.31	1315.06	-0.38	0.40	-0.39	0.04	0.44
27	1323.54	-	1318.04	1322.34	1311.27	1314.47	1314.83	1320.61	-0.13	0.47	-0.66	-0.24	0.27
28	2560.06	-	2530.46	2545.93	2519.35	2538.87	2518.14	2548.74	0.11	0.39	-1.64	-0.49	-0.05
29	2191.46	-	2173.02	2194.69	2166.14	2170.47	2170.09	2198.76	0.19	1.30	-0.98	-0.13	0.18
30	1775.45	-	1760.59	1766.02	1746.82	1753.78	1751.78	1765.16	-0.05	0.65	-1.33	-0.50	0.28
31	2282.28	-	2244.13	2254.72	2227.79	2240.73	2232.15	2253.74	-0.04	0.58	-2.20	-0.53	0.20
32	2233.27	-	2196.85	2208.16	2177.66	2190.15	2180.85	2204.18	-0.18	0.64	-2.35	-0.73	0.15
33	2284.82	-	2261.68	2276.19	2239.91	2252.03	2247.51	2271.08	-0.22	0.85	-1.63	-0.63	0.34
34	1191.13	-	1157.22	1161.62	1147.67	1153.78	1152.91	1162.09	0.04	0.72	-3.21	-0.37	0.46
35	1435.23	-	1401.17	1409.01	1388.55	1396.35	1391.35	1400.58	-0.60	0.30	-3.06	-0.70	0.20
36	1729.79	-	1669.44	1682.89	1656.00	1665.01	1651.34	1662.71	-1.20	-0.14	-4.54	-1.08	-0.28
avg									*-0.14*	*0.22*	*-0.81*	*-0.21*	*0.09*
time (min)	*24.2*		*14.2*		*14.5*		*6.2*						

bst: best solution score over ten runs, **avg:** average solution score over ten runs. The ***bst*** and ***avg*** values are averages over Classes 2 – 5.
AVG and ***BST*** column groups refer to percent gaps relatively to the ***avg*** and ***bst*** values, respectively.
g_{GRASP} = 100 (*LS-2LSPD - GRASP*)/*GRASP*, g_{PRMP} = 100 (*LS-2LSPD – PRMP*)/*PRMP*, g_{VNS} = 100 (*LS-2LSPD – VNS*)/*VNS*

Our algorithm scores (both best and average solution values) are compared to previously published methodologies in Table 1, where averaged values over Classes 2-5 are reported. The algorithms considered are: (1) the GRASP approach of Duhamel et al. (2011) – AMD Opteron (2.1 GHz), (2) the PRMP metaheuristic of Zachariadis et al. (2013) - Intel Core 2 Duo E6600 (2.4 GHz), and (3) the VNS method of Wei et al. (2015) - Intel Xeon E5430 (2.66 GHz). Note that the results of Dominguez et al. (2014) are not provided, because the authors have only solved a limited subset of the benchmark instances for the 2|UO|L configuration. We observe that our solution method exhibits a very competitive and robust performance. In terms of the average solution scores over ten runs, LS-2LSPD improves our previous metaheuristic (PRMP) by 0.14%, whereas the VNS methodology produces solution values which are on average 0.22% better than the LS-2L-SPD ones. Regarding the best solution values obtained over the ten algorithmic runs, LS-2LSPD improves both the GRASP and PRMP scores by 0.81% and 0.21%, respectively. The VNS methodology produces slightly better best solutions (by 0.09% on average). Concerning the computational times, it is not our intention to perform a detailed comparison because the presented algorithms have been executed on very different conditions (processors, OS, RAM, programming languages etc.). Briefly and regarding the two most effective algorithms, LS-2LSPD and VNS appear to require comparable computational effort. The LS-2LSPD method generated the final solutions in shorter computational periods than the VNS method (LS-2LSPD: 6.2 min, VNS: 14.5 min). However, the LS-2LSPD runs were performed on a faster processor (LS-2LSPD: Intel Xeon E5-2650 v2 – 2.6 GHz, VNS: Intel Xeon E5430 - 2.66 GHz).

7.3.2. Results on the 2|UR|L configuration of 2L-CVRP

Each of the 144 instances was solved 10 times under the 2|UR|L configuration. The results are summarized in Table A.5. Our algorithm exhibits a rather stable performance, as the average gap between the best and average solution scores over the ten runs is limited to 0.33%, 0.39%, 0.38% and 0.34% for Classes 2, 3, 4 and 5, respectively. The computational times required for obtaining the final solution of each run are slightly increased compared to the configuration that does not accept item rotations. On average, the CPU time elapsed when the final solution of each run was generated is equal to 347.0 sec, 364.2 sec, 469.5 and 473.7 sec for Classes 2, 3, 4 and 5, respectively.

The best LS-2LSPD scores are compared against the best known solution values in Table A.6. We observe that LS-2LSPD managed to match or improve the majority of the previously best known 2|UR|L solution scores. More specifically, our method improves 79 and matches 58 BKS scores, respectively. Note that some minor reported discrepancies ($\pm$0.01) may be caused by different rounding schemes. The average BKS improvement is equal to 0.75% (Class 2: 0.42%, Class 3: 0.72%, Class 4: 1.11%, Class 5: 0.75%), ranging up to a significant 4.67% for test problem 28 of Class 4.

In addition, to compare our algorithm with previously published methods for cases where item rotations are allowed, we provide Table 2. Our method is compared against the Ant Colony Optimization (ACO) algorithm of Fuellerer et al. (2009) and the multi-start biased-randomized (MS-BR) method of Dominguez et al. (2014). The ACO values are available at http://prolog.univie.ac.at/research/VRPandBPP/. They were obtained by the 3-hr ACO run which was executed on an Intel Pentium IV, 3.2 GHz. The MS-BR algorithm was executed on an Intel Core 2 (2.4 GHz) processor. The scores are reported in the article Dominguez et al. (2014). Regarding the average solution values over the ten runs, we see that the proposed method improves the ACO and MS-BR ones

by 1.42% and 0.77%, respectively. The same applies for the best solution values obtained after ten algorithmic runs: our methodology produces solutions which improve the ones generated by ACO and MS-BR by 1.39% and 0.80% on average, respectively. As far as the computational times are concerned, no secure comparisons among the methods can be conducted, due to the completely different execution conditions. Briefly, the MS-BR appears to be the fastest method. Although MS-BR was executed on a slower processor than the one used for the LS-2LPD algorithm, the average time elapsed for obtaining the final solutions is almost half the time required by LS-2LPD (MS-BR: 3.6 min, LS-2LPD: 6.9 min). Regarding the ACO method, it was executed on a much slower processor, however on average it appears to require almost ten times the time required by the MS-BR method (MS-BR: 3.6 min, ACO: 37.0 min).

Table 2. Comparison of the best performing algorithms for the 2|UR|L version of 2L-CVRP

	ACO		MS-BR		LS-2LSPD		BST		AVG	
Instance	**bst**	**avg**	**bst**	**avg**	**bst**	**avg**	g_{ACO}	$g_{MS\text{-}BR}$	g_{ACO}	$g_{MS\text{-}BR}$
1	281.16	281.26	280.84	281.14	281.13	281.13	-0.01	0.10	-0.05	-0.01
2	341.02	341.02	339.26	339.26	339.26	339.26	-0.52	0.00	-0.52	0.00
3	372.93	374.35	370.62	371.59	372.86	372.86	-0.02	0.60	-0.40	0.34
4	435.00	435.01	435.00	435.00	435.01	435.00	0.00	0.00	0.00	0.00
5	378.60	378.60	378.59	378.59	378.60	378.59	0.00	0.00	0.00	0.00
6	497.05	497.10	497.05	497.05	497.05	497.05	0.00	0.00	-0.01	0.00
7	688.50	688.50	681.00	681.00	681.00	681.00	-1.09	0.00	-1.09	0.00
8	678.75	679.09	675.46	676.51	675.46	676.49	-0.49	0.00	-0.38	0.00
9	612.02	612.19	612.01	612.01	612.01	612.01	0.00	0.00	-0.03	0.00
10	671.00	673.20	667.65	668.68	660.22	660.62	-1.61	-1.11	-1.87	-1.21
11	698.25	702.46	690.56	693.19	690.98	690.98	-1.04	0.06	-1.63	-0.32
12	611.12	613.11	611.06	611.54	610.06	611.06	-0.17	-0.16	-0.33	-0.08
13	2468.19	2468.49	2437.15	2446.34	2428.87	2428.87	-1.59	-0.34	-1.61	-0.71
14	974.81	980.78	968.55	975.90	964.39	965.56	-1.07	-0.43	-1.55	-1.06
15	1132.49	1143.07	1112.00	1123.13	1110.60	1111.79	-1.93	-0.13	-2.74	-1.01
16	699.80	699.79	699.80	699.80	699.80	699.80	0.00	0.00	0.00	0.00
17	862.37	862.69	861.79	861.85	861.79	861.80	-0.07	0.00	-0.10	-0.01
18	1012.20	1013.32	999.22	1004.90	1002.63	1002.92	-0.94	0.34	-1.03	-0.20
19	726.96	730.17	722.17	724.38	715.79	719.09	-1.54	-0.88	-1.52	-0.73
20	508.69	509.93	501.90	504.80	500.95	501.11	-1.52	-0.19	-1.73	-0.73
21	989.24	994.16	977.03	983.31	969.82	973.51	-1.96	-0.74	-2.08	-1.00
22	1008.52	1011.82	1001.75	1004.02	998.67	1000.99	-0.98	-0.31	-1.07	-0.30
23	1024.25	1030.88	1011.19	1014.45	1002.79	1005.70	-2.10	-0.83	-2.44	-0.86
24	1098.60	1103.07	1092.90	1095.09	1083.72	1089.03	-1.35	-0.84	-1.27	-0.55
25	1323.84	1331.31	1320.27	1325.79	1299.68	1306.44	-1.82	-1.56	-1.87	-1.46
26	1314.34	1320.57	1302.52	1310.99	1289.84	1299.77	-1.86	-0.97	-1.58	-0.86
27	1309.76	1316.31	1304.14	1307.82	1281.28	1292.31	-2.17	-1.75	-1.82	-1.19
28	2526.81	2561.56	2518.51	2525.12	2457.50	2480.36	-2.74	-2.42	-3.17	-1.77
29	2175.33	2188.84	2161.43	2166.34	2128.75	2159.21	-2.14	-1.51	-1.35	-0.33
30	1742.15	1756.13	1742.01	1747.21	1707.66	1719.76	-1.98	-1.97	-2.07	-1.57
31	2227.74	2240.44	2204.44	2221.54	2164.38	2180.97	-2.84	-1.82	-2.65	-1.83
32	2180.18	2193.73	2167.61	2179.89	2127.70	2147.19	-2.41	-1.84	-2.12	-1.50
33	2239.04	2251.86	2222.42	2232.53	2185.14	2206.30	-2.41	-1.68	-2.02	-1.17
34	1149.87	1157.32	1142.25	1149.95	1117.18	1127.24	-2.84	-2.19	-2.60	-1.97
35	1387.45	1393.37	1392.05	1399.92	1349.66	1360.29	-2.72	-3.05	-2.37	-2.83
36	1670.67	1682.00	1653.05	1662.05	1599.28	1612.53	-4.27	-3.25	-4.13	-2.98
avg							*-1.39*	*-0.80*	*-1.42*	*-0.77*
time (min)	*37.0*		*3.6*		*6.9*					

bst: best solution score over ten runs, **avg:** average solution score over ten runs. The **bst** and **avg** values are averages over Classes 2 – 5.
AVG and **BST** column groups refer to percent gaps relatively to the **avg** and **bst** values, respectively.
g_{ACO}: 100 (*LS-2LSPD - ACO*)/*ACO*, $g_{MS\text{-}BR}$: 100 (*LS-LD – MS-BR*)/*MS-BR*

7.4. Computational Results on the VRPSPD model

To further examine the effectiveness of the employed local-search operators, we have applied *LS-2LSPD* on the well-studied VRPSPD instances of Tang-Montanè & Galvão (2006). These instances can be regarded a special case of 2L-SPD instances for which the loading constraints are always satisfied. Specifically, we have solved the instance set involving up to 200 customers. The average and best results after ten executions of our algorithm are reported in Table 3. Moreover, Table 3 compares our best scores against the best known solutions published

for these instances. Our algorithm exhibited a very stable behavior: the percent gaps between the average and best solution scores for each instance ranged up to 0.78%. In addition, it managed to match five out of the 12 best known solution scores (taken from Subramanian et al. (2013b)). The average gap between our best and the best known solution scores was limited to 0.18%. Bearing in mind that our routing component was mainly designed for speed, in order to efficiently integrate the computationally expensive loading feasibility procedures, the performance of the LS-2LSPD method on the VRPSPD instances is satisfactory.

Table 3. Computational Results on the VRPSPD model

Instance	n	avg	bst	%avg	BKS	%bst
r101	100	1009.95	**1009.95**	0.00	**1009.95**	0.00
r201	100	666.20	**666.20**	0.00	**666.20**	0.00
c101	100	1220.99	1220.99	0.00	**1220.18**	0.07
c201	100	662.07	**662.07**	0.00	**662.07**	0.00
rc101	100	1059.32	**1059.32**	0.00	**1059.32**	0.00
rc201	100	672.92	**672.92**	0.00	**672.92**	0.00
R1_2_1	200	3399.93	3376.30	0.70	**3353.80**	0.67
R2_2_1	200	1671.25	1666.09	0.31	**1665.58**	0.03
C1_2_1	200	3657.14	3652.76	0.12	**3628.51**	0.67
C2_2_1	200	1729.55	1728.34	0.07	**1726.59**	0.10
RC1_2_1	200	3349.55	3323.56	0.78	**3303.70**	0.60
RC2_2_1	200	1568.72	1560.51	0.53	**1560.00**	0.03
average				*0.21*		*0.18*

n: number of customers, **avg:** average objective score over ten runs, **bst:** best solution score obtained over ten runs, **%avg**: the percent gap between the bst and avg scores (=100(avg-bst)/bst), **BKS**: the best known solution score, **%bst**: the percent gap between the bst and BKS scores (=100(bst-BKS)/BKS).

7.5. Computational Results on the 2L-SPD model

We deal with two versions of the 2L-SPD model: the first one does not allow item rotations (2|O|SPD), whereas the second one considers that items may be rotated by 90° (2|R|SPD).

7.5.1. Results on the 2|O|SPD version of the 2L-SPD model

Each of the 144 2L-SPD benchmark instances reported in Table A.1 was solved 10 times. Items were assumed to be loaded with fixed orientation as dictated by the 2|O|SPD model version. The results obtained are summarized in Table 4. From the results obtained, we see that the proposed methodology was rather stable in terms of the quality of the solutions generated in each run. The percent gaps between the average and highest quality solution were limited to 1.39%, 1.17%, 1.14% and 1.04%, for Classes 2, 3, 4 and 5 respectively. Regarding the computational effort required, LS-2LSPD managed to obtain the final solution of each run in reasonable run times, taking into account both the challenging loading constraints of 2L-SPD and the rather large scale of the examined test cases. For Classes 2, 3 4 and 5 the average computational times ranged up to 101.9, 221.8, 196.5 and 259.0 CPU minutes, respectively. As an overall comment, the computational effort strongly depends on the scale of the instance and on the total number of items per travelling arc. Moreover, it depends on the relation between the weight (1-d) and loading (2-d) constraints: the tighter the 1-d constraints (which are examined in constant time), the more routes declared weight infeasible, and thus the fewer routes to be examined by the time consuming loading feasibility procedures. Another factor which has a strong impact on the required computational effort is associated with the tightness of the loading constraints: the tighter the loading constraints, the more unproductive packing attempts are performed by the proposed heuristic (§5.3).

Table 4. Summary of the results obtained for the 2|O|SPD version of 2L-SPD

Inst	Class 2				Class 3				Class 4				Class 5			
	bst	avg	t	%g	bst	avg	t	%g	bst	avg	t	%g	bst	avg	t	%g
1	238.67	238.67	0.0	0.00	238.67	238.67	0.0	0.00	252.15	252.15	4.1	0.00	244.10	244.10	1.8	0.00
2	266.80	266.80	0.0	0.00	270.97	270.97	0.0	0.00	240.78	240.78	9.7	0.00	238.64	238.64	4.5	0.00
3	325.29	325.29	0.2	0.00	323.22	323.22	4.1	0.00	299.65	299.65	58.3	0.00	283.91	283.91	8.3	0.00
4	353.21	353.21	0.2	0.00	312.91	312.91	5.6	0.00	358.69	358.69	8.4	0.00	290.91	290.91	7.2	0.00
5	319.13	319.13	4.0	0.00	289.88	289.88	9.7	0.00	340.72	340.72	2.1	0.00	301.38	301.38	56.1	0.00
6	340.64	340.64	3.1	0.00	339.71	339.71	42.0	0.00	374.84	374.84	0.8	0.00	289.88	289.88	33.6	0.00
7	593.59	593.59	2.5	0.00	531.39	531.39	134.4	0.00	590.33	590.33	2.7	0.00	595.43	595.43	50.2	0.00
8	621.55	621.55	4.8	0.00	614.34	614.34	2.6	0.00	588.74	588.74	6.0	0.00	554.14	554.14	190.6	0.00
9	519.85	529.77	23.8	1.91	380.31	384.62	207.7	1.13	392.69	395.01	129.6	0.59	364.57	364.57	130.0	0.00
10	545.44	545.44	143.5	0.00	534.30	534.30	8.0	0.00	537.47	537.49	263.3	0.01	543.90	543.90	97.4	0.00
11	564.83	564.83	31.1	0.00	557.21	557.21	23.5	0.00	553.90	553.90	48.1	0.00	461.15	461.15	38.6	0.00
12	441.82	441.82	18.3	0.00	365.71	365.71	144.1	0.00	433.15	433.15	71.2	0.00	370.39	370.39	241.6	0.00
13	2090.57	2090.57	6.0	0.00	2000.80	2000.80	66.3	0.00	2113.89	2120.95	648.2	0.33	1968.93	1985.38	885.9	0.84
14	695.88	695.88	974.4	0.00	688.65	688.65	435.2	0.00	682.20	682.20	13.1	0.00	680.69	680.69	20.0	0.00
15	693.00	693.00	941.3	0.00	708.28	721.76	883.1	1.90	811.76	824.21	125.6	1.53	818.96	818.96	542.4	0.00
16	518.41	518.41	35.9	0.00	424.95	424.95	76.7	0.00	442.75	445.58	423.4	0.64	414.02	414.02	123.5	0.00
17	465.67	465.67	79.0	0.00	438.22	438.99	587.7	0.18	469.97	470.01	485.7	0.01	414.33	416.56	684.6	0.54
18	790.78	802.58	5630.8	1.49	796.70	797.15	379.3	0.06	881.94	886.97	1619.7	0.57	749.49	750.40	871.7	0.12
19	572.61	585.84	2413.6	2.31	570.90	572.63	274.9	0.30	574.72	577.00	2159.7	0.40	500.39	501.64	6090.6	0.25
20	363.01	365.02	863.7	0.55	350.87	352.13	1163.8	0.36	374.09	374.83	1070.4	0.20	326.34	328.22	3640.6	0.58
21	748.99	750.08	928.3	0.15	802.08	805.13	533.7	0.38	732.68	745.26	550.7	1.72	674.92	684.79	5089.0	1.46
22	776.03	793.13	2801.1	2.20	768.71	784.93	2493.5	2.11	775.10	782.55	1762.5	0.96	697.32	701.54	3375.6	0.60
23	776.75	794.49	1334.5	2.28	776.69	789.39	1242.8	1.64	768.40	785.48	1900.4	2.22	705.21	708.30	4833.5	0.44
24	863.22	874.01	969.4	1.25	793.29	814.08	1374.0	2.62	809.93	818.49	1531.9	1.06	768.26	773.48	680.8	0.68
25	997.39	1009.23	4040.1	1.19	962.60	980.83	2034.3	1.89	978.61	992.83	3700.4	1.45	932.30	943.16	1710.4	1.17
26	923.44	942.22	2144.4	2.03	894.47	912.64	1197.5	2.03	961.67	970.19	1150.3	0.89	820.32	827.44	2985.3	0.87
27	950.19	984.68	2399.4	3.63	972.03	995.79	2446.5	2.44	959.36	1000.70	7621.4	4.31	880.34	903.01	8650.9	2.57
28	1734.38	1801.79	5603.4	3.89	1732.61	1812.25	2586.1	4.60	1611.32	1681.64	9936.8	4.36	1445.43	1522.87	9924.5	5.36
29	1442.45	1528.46	5893.6	5.96	1425.99	1507.76	13308.1	5.73	1457.94	1535.83	11789.9	5.34	1460.64	1542.87	15539.9	5.63
30	1252.24	1297.60	3712.9	3.62	1265.23	1292.04	3032.2	2.12	1240.26	1260.72	2798.0	1.65	1049.86	1081.51	5725.0	3.02
31	1524.08	1596.65	3930.7	4.76	1518.89	1560.41	3043.3	2.73	1575.42	1624.96	3393.6	3.14	1365.49	1388.53	6816.5	1.69
32	1527.37	1580.13	3629.2	3.45	1492.44	1525.01	3207.0	2.18	1506.75	1553.23	4167.5	3.08	1322.59	1370.44	7399.8	3.62
33	1512.39	1570.78	2093.8	3.86	1539.63	1591.89	3266.7	3.39	1529.56	1581.49	3070.0	3.40	1378.72	1430.10	7958.2	3.73
34	769.19	783.33	4921.7	1.84	793.21	811.60	4327.3	2.32	784.74	792.41	5172.9	0.98	691.84	697.69	8004.3	0.85
35	901.41	915.08	3065.0	1.52	923.19	936.82	5472.1	1.48	960.54	975.52	7975.5	1.56	811.36	831.40	10124.4	2.47
36	1028.41	1050.37	6114.0	2.14	1053.96	1058.46	11151.3	0.43	994.88	1002.94	8361.2	0.81	906.68	915.11	14647.6	0.93
avg			*1798.8*	*1.39*			*1810.1*	*1.17*			*2278.7*	*1.14*			*3532.9*	*1.04*

bst: the best solution score obtained over the ten runs, avg: the average solution score over the ten runs, **t**: the average computational time elapsed for obtaining the final solution over the ten runs (in CPU sec), **%g:** the percent deviation between our best and average scores (= 100(avg-bst)/bst)

Table A.7 provides further details on the best solution obtained for each instance. More specifically, it reports the average and maximum loading area utilization of the solution arcs. In addition, another important metric is reported illustrating the tightness of the loading constraints: This is the ratio of the feasible routes to the total number of routes which were examined in terms of the loading constraints. This metric, denoted as T_RHL, has been obtained as the total number of true values contained in the hashtable *RHL* divided by the total *RHL* entries (of the run that yielded the best solution score). We see that the average loading area utilization over the solution arcs stays on significant levels: Class 2: 59.1%, Class 3: 66.9%, Class 4: 67.1%, and Class 5: 71.0%, whereas the maximum loading space utilization goes up to 89.8%, 93.4%, 93.8% and 96.0%, for Classes 2, 3, 4 and 5, respectively. These high utilization levels indicate that our loading examination method is much effective and that the loading constraints of the 2L-SPD test-beds are challenging which in turn implies that they were appropriately designed to act as a performance comparison basis for 2L-SPD algorithms. Finally, the T_RHL ratios are widely distributed within [0, 100]. The T-RHL values ranged between [3.4%, 83.5%], [2.4%, 85.1%], [3.0%, 86.1%] and [4.2%, 87.6%] for Classes 2, 3, 4 and 5, respectively.

7.5.2. Results on the 2|R|SPD version of the 2L-SPD model

Each of the 144 2L-SPD benchmark problems was solved 10 times under the 2|R|SPD version. The results obtained are summarized in Table 5. The proposed algorithm was again rather stable: the percent gaps between

the average and best solution score achieved for each instance averaged at 1.07%, 0.94%, 0.88% and 0.99% for Classes 2, 3, 4 and 5, respectively. These gaps are slightly lower than the ones observed for the fixed orientation version (Class 2: 1.39%, 1.17%, 1.14%, Class 5: 1.04%). This can be attributed to the fact that the relaxed loading constraints form a smoother solution space. The average computational time for obtaining the final solution of each run ranged up to 105.5, 133.2, 226.5 and 362.8 CPU minutes, for Classes 2, 3, 4 and 5, respectively. On average, the computational times for the 2|R|SPD version were lower than those recorded for the 2|O|SPD problem version. This can be attributed to the fact that the relaxation of the loading constraints reduced the number of unproductive calls to the packing heuristic method (see the increased values of the T-RHL column in Table A.8).

Table 5. Summary of the results obtained for the 2|R|SPD version of 2L-SPD

Inst	Class 2				Class 3				Class 4				Class 5			
	bst	avg	t	%g	bst	avg	t	%g	bst	avg	t	%g	bst	avg	t	%g
1	238.67	238.67	1.7	0.00	238.67	238.67	2.1	0.00	251.32	251.32	4.5	0.00	244.10	244.10	4.2	0.00
2	266.80	266.80	0.3	0.00	269.77	269.77	5.7	0.00	240.78	240.78	4.1	0.00	238.64	238.64	3.2	0.00
3	305.47	305.47	2.8	0.00	312.61	312.61	10.7	0.00	299.65	299.65	22.8	0.00	283.91	283.91	22.5	0.00
4	353.21	353.21	2.0	0.00	309.77	309.77	1.4	0.00	358.01	358.01	11.2	0.00	290.91	290.91	94.6	0.00
5	317.94	317.94	6.5	0.00	289.88	289.88	8.6	0.00	340.72	340.72	12.4	0.00	289.88	289.88	30.1	0.00
6	325.54	325.54	8.3	0.00	326.68	326.68	5.8	0.00	374.84	374.84	13.1	0.00	289.88	289.88	12.6	0.00
7	588.74	588.74	6.7	0.00	503.34	503.34	4.8	0.00	588.88	588.88	59.4	0.00	595.43	595.43	19.1	0.00
8	595.11	595.11	22.9	0.00	609.96	609.96	11.7	0.00	588.74	588.74	17.4	0.00	546.91	549.62	76.3	0.50
9	510.27	523.56	6.6	2.60	375.38	375.38	171.4	0.00	380.73	380.73	219.6	0.00	364.57	364.57	20.4	0.00
10	542.13	542.13	73.0	0.00	479.12	483.99	13.0	1.02	533.04	537.49	99.6	0.84	543.90	543.90	41.2	0.00
11	558.02	558.02	43.5	0.00	556.46	556.46	48.7	0.00	552.23	552.37	77.1	0.03	461.15	461.15	19.5	0.00
12	437.53	437.53	1.6	0.00	363.56	363.56	435.0	0.00	433.15	435.22	48.6	0.48	368.98	369.31	588.6	0.09
13	2068.81	2078.12	142.4	0.45	1971.92	1971.92	72.2	0.00	2050.08	2065.67	377.0	0.76	1968.93	1968.93	614.1	0.00
14	686.88	686.88	16.8	0.00	686.69	686.87	725.2	0.03	682.20	682.20	19.0	0.00	655.07	659.03	1080.3	0.60
15	686.53	686.53	183.9	0.00	692.51	694.24	94.7	0.25	789.21	789.48	636.1	0.03	817.86	817.86	196.9	0.00
16	518.41	520.53	22.0	0.41	419.55	420.06	115.5	0.12	442.75	443.14	239.3	0.09	414.02	414.02	437.5	0.00
17	455.84	456.52	135.5	0.15	435.44	435.53	239.8	0.02	461.92	462.77	450.1	0.19	413.27	413.66	239.7	0.10
18	782.35	782.79	934.2	0.06	776.84	778.63	282.6	0.23	864.79	871.15	1436.0	0.74	732.84	744.31	210.7	1.56
19	544.60	547.07	928.7	0.45	562.95	563.85	381.6	0.16	565.54	565.56	1328.4	0.00	495.10	497.30	3292.2	0.45
20	348.78	350.44	969.9	0.48	348.13	349.82	1066.9	0.48	365.06	365.43	1333.0	0.10	316.76	326.12	2328.6	2.95
21	723.48	727.04	736.0	0.49	770.07	775.90	1211.3	0.76	719.54	726.70	764.3	0.99	668.91	678.28	2715.8	1.40
22	737.58	739.04	1007.0	0.20	750.04	755.04	1442.1	0.67	772.20	776.36	1318.9	0.54	692.53	694.34	3631.1	0.26
23	727.18	741.41	575.6	1.96	755.10	764.95	1409.1	1.31	757.06	760.95	2713.8	0.51	702.13	702.23	3610.7	0.01
24	826.85	839.85	682.0	1.57	773.96	804.31	999.5	3.92	807.05	811.84	1486.8	0.59	767.39	771.44	962.6	0.53
25	946.82	962.60	1573.2	1.67	936.84	962.11	3008.9	2.70	958.21	971.98	3829.2	1.44	932.30	942.18	958.7	1.06
26	860.75	877.75	879.7	1.98	887.17	889.46	1487.2	0.26	919.26	924.01	2169.1	0.52	813.10	823.53	1880.5	1.28
27	915.87	941.86	1307.8	2.84	955.16	966.12	2671.1	1.15	937.74	956.38	5743.1	1.99	876.90	890.88	4960.7	1.59
28	1636.11	1715.23	3191.6	4.84	1646.17	1714.83	3475.4	4.17	1574.10	1653.99	5897.1	5.07	1410.59	1489.10	6689.0	5.57
29	1376.34	1435.80	5689.6	4.32	1400.30	1469.30	6729.5	4.93	1444.90	1506.19	10154.2	4.24	1421.87	1502.47	14108.9	5.67
30	1190.07	1211.45	2816.2	1.80	1219.56	1244.26	3096.1	2.03	1209.14	1227.25	5773.8	1.50	1035.35	1058.82	3947.3	2.27
31	1475.13	1508.05	3077.7	2.23	1479.90	1511.82	2672.6	2.16	1531.47	1579.08	5466.4	3.11	1343.37	1364.13	5928.1	1.55
32	1444.64	1495.24	2510.4	3.50	1438.81	1476.43	2700.3	2.61	1465.22	1499.76	6304.7	2.36	1311.35	1337.64	5828.1	2.00
33	1443.18	1481.65	3112.0	2.67	1499.28	1532.34	2921.1	2.20	1498.63	1534.78	4886.7	2.41	1345.28	1373.10	6045.7	2.07
34	735.76	743.06	3959.1	0.99	773.89	781.16	2485.2	0.94	765.13	773.50	5088.7	1.09	684.73	691.74	6573.5	1.02
35	857.47	870.22	3795.9	1.49	893.43	905.31	3766.5	1.33	924.73	935.98	8185.5	1.22	798.29	814.12	8028.4	1.98
36	977.71	991.64	6329.6	1.42	1012.00	1017.51	7993.8	0.54	969.51	979.48	13592.2	1.03	897.14	906.99	21768.9	1.10
avg			*1243.1*	*1.07*			*1438.0*	*0.94*			*2494.0*	*0.88*			*2971.4*	*0.99*

The notation of Table 4 is used

To gain more insight on the solution structures obtained for the 2|R|SPD version of 2L-SPD, we provide Table A.8. It contains the average and maximal loading area utilization in the best solution of every test problem. We can see that the average loading area utilization for Classes 2, 3, 4 and 5 went up to 63.1%, 68.8%, 69.6% and 71.6% respectively (for the 2|O|SPD version, the average utilization is: 59.1%, 66.9%, 67.1%, 71.0%, for Classes 2, 3, 4 and 5, respectively). The same picture is seen for the maximal loading area utilization: for the 2|R|SPD version, it is equal to 93.2%, 96.9%, 96.5% and 97.7% for Classes 2,3, 4 and 5, respectively, while the corresponding values for the 2|O|SPD version are 89.8%, 93.4%, 93.8% and 96%.0. We can see that the relaxation of the loading constraint (rotations allowed) has a stronger impact for Class 2 which involves fewer and larger transported items. A final comment is related to the ratio of the routes found to be loading feasible compared to the total routes examined regarding the loading feasibility, denoted as T_RHL: we observe that for

the 2|R|SPD model version and for Classes 2, 3, 4 and 5 the T-RHL values go up to 34.5%, 32.2%, 31.3% and 37.2%, respectively. For the 2|O|SPD version, the corresponding T_RHL averages are 28.5%, 27.4%, 28.2% and 35.7%. Again, we conclude that allowing item rotation plays a more significant role for the test problems of Classes 2 and 3 which involve fewer and larger transported items.

Table 6. Summary of the results obtained for the instances introduced for the LIFO version of 2L-SPD

Instance	Fixed Item Orientation					Item Rotations				
	2|O|SPD-L		2|O|SPD			2|R|SPD-L		2|R|SPD		
	z	k	z	k	%g	z	k	z	k	%g
L01-4	284.42	3	258.11	3	10.19	281.66	3	258.11	3	9.12
L02-4	285.57	3	263.91	3	8.21	285.57	3	260.22	3	9.74
L03-4	350.98	4	332.28	3	5.63	350.98	4	328.30	3	6.91
L04-4	360.13	4	319.56	3	12.70	358.20	4	311.90	3	14.84
L05-4	383.86	4	339.44	4	13.09	378.23	4	339.44	4	11.43
L06-4	391.83	4	354.43	4	10.55	380.95	4	354.43	4	7.48
L07-4	601.83	3	527.64	2	14.06	601.83	3	527.64	2	14.06
L08-4	618.40	3	583.94	3	5.90	616.89	3	583.94	3	5.64
L09-4	437.24	5	400.62	4	9.14	427.86	4	400.62	4	6.80
L10-4	599.61	4	524.49	4	14.32	597.14	4	524.49	4	13.85
L11-4	677.46	5	629.69	4	7.59	675.16	5	629.69	4	7.22
L12-4	394.15	4	356.78	3	10.47	392.06	4	356.78	3	9.89
L13-4	2311.91	4	2005.51	3	15.28	2246.00	4	2005.51	3	11.99
L14-4	699.56	3	678.47	3	3.11	693.34	3	678.47	3	2.19
L15-4	917.42	4	838.51	4	9.41	889.04	4	838.51	4	6.03
L16-4	440.79	4	403.90	3	9.13	433.45	4	397.51	3	9.04
L17-4	473.57	5	410.27	4	15.43	458.04	5	410.27	4	11.64
L18-4	981.21	6	860.64	6	14.01	942.45	6	857.94	6	9.85
L19-4	624.63	7	551.93	6	13.17	619.71	7	549.83	6	12.71
L20-4	462.16	10	382.72	8	20.76	438.09	9	378.34	8	15.79
L21-4	824.20	8	716.40	7	15.05	804.06	8	714.54	7	12.53
L22-4	860.65	10	754.27	8	14.10	853.20	10	750.14	8	13.74
L23-4	858.69	10	705.20	7	21.77	820.55	9	698.93	7	17.40
L24-4	821.03	10	710.84	7	15.50	799.38	10	709.66	7	12.64
L25-4	1073.55	13	900.47	9	19.22	1043.15	12	893.81	9	16.71
L26-4	1077.06	13	824.22	10	30.68	1028.96	12	818.63	9	25.69
L01-5	273.61	3	245.39	2	11.50	272.03	3	245.39	2	10.86
L02-5	269.77	3	257.86	3	4.62	264.96	2	257.86	3	2.75
L03-5	356.54	3	305.44	3	16.73	335.19	3	305.44	3	9.74
L04-5	365.46	4	336.28	4	8.68	364.29	4	321.16	4	13.43
L05-5	365.19	3	319.13	3	14.43	362.12	3	319.13	3	13.47
L06-5	397.74	4	351.30	3	13.22	388.31	4	351.30	3	10.54
L07-5	613.67	3	563.25	3	8.95	607.70	3	563.25	3	7.89
L08-5	634.86	3	530.96	2	19.57	617.22	3	530.96	2	16.25
L09-5	462.53	4	413.03	4	11.98	436.32	4	413.03	4	5.64
L10-5	561.70	4	504.79	3	11.27	558.44	4	503.17	3	10.98
L11-5	546.36	4	497.53	3	9.81	519.06	4	497.53	3	4.33
L12-5	399.25	4	360.92	3	10.62	396.22	4	360.92	3	9.78
L13-5	2301.19	5	2111.04	4	9.01	2285.59	4	2111.04	4	8.27
L14-5	775.26	3	683.98	3	13.35	750.11	3	683.98	3	9.67
L15-5	904.09	4	840.39	4	7.58	894.47	4	840.39	4	6.44
L16-5	455.51	5	422.23	4	7.88	455.51	5	422.04	4	7.93
L17-5	520.39	5	446.51	4	16.55	505.94	5	443.84	4	13.99
L18-5	861.81	5	792.60	5	8.73	859.93	5	786.42	5	9.35
L19-5	584.11	6	524.27	5	11.41	578.20	6	524.27	5	10.29
L20-5	389.85	9	360.59	8	8.11	386.18	9	359.14	8	7.53
L21-5	752.77	8	655.97	6	14.76	748.35	8	655.97	6	14.08
L22-5	810.75	9	685.02	6	18.35	799.98	8	680.04	6	17.64
L23-5	781.51	8	686.04	7	13.92	768.16	8	685.58	7	12.05
L24-5	780.63	8	692.38	7	12.75	777.19	8	691.99	6	12.31
L25-5	979.99	11	844.53	8	16.04	971.98	11	844.53	8	15.09
L26-5	952.87	11	789.18	8	20.74	928.57	11	782.35	8	18.69
avg		*5.62*		*4.62*	*12.67*		*5.46*		*4.58*	*11.04*
CPU time (min)	*74.4*		*3.2*			*57.9*		*2.5*		

z: best objective score obtained for each instance, ***k***: number of vehicles used in the best solution, ***%g***: the percent gap between the best scores obtained with and without the LIFO constraints (= 100 (LIFO – noLIFO) / LIFO)

7.6. Computational Results on the 2L-SPD model with LIFO constraints

As already stated, we examine two LIFO 2L-SPD model configurations, depending on whether item rotations are allowed: 2|O|SPD-L (fixed orientation) and 2|R|SPD-L (rotations allowed). The benchmark instances introduced in §7.2 were solved under both the LIFO version of 2L-SPD and the basic 2L-SPD model, in order to gain insight on the impact of the LIFO constraints on the routing objective and the vehicle usage. Our algorithm was applied ten times on each instance. The analytic solutions for these runs are reported in Tables A.9 and A.10 for the 2L-SPD with and without LIFO constraints, respectively. In Table 6, the best solution scores for each model are provided. We observe that the consideration of LIFO constraints significantly increases the routing costs and the number of vehicles used: on average, the routing objective is augmented by 12.67% when items are inserted with fixed orientation. This average gap is 11.04% when item rotations are allowed. In terms of the vehicle usage, the 2|O|SPD-L solutions require on average 5.62 routes, while under the 2|O|SPD configuration the average number of routes goes down to 4.62. For the 2|R|SPD-L and 2|R|SPD models, the average vehicle utilization is 5.46 and 4.58, respectively. The obtained results indicate that dropping the LIFO constraints from the basic 2L-SPD model can greatly reduce the routing costs involved. This can compensate for the extra effort required for the necessary item rearrangements.

From Table 6, we observe that the best 2L-SPD solutions were generated in shorter computational times compared to the ones required for obtaining the best solutions for the LIFO version of the model. This observation is contrary to what one would expect: the basic 2L-SPD model requires the determination of feasible item packings for all travelled arcs, whereas under the 2L-SPD with LIFO constraints, only two packings have to be determined for each vehicle route (one for the pick-up and one for the delivery items). Obviously, the computational time required for the two model configurations depend on various parameters, such as the tightness of the constraints (which in turn have an effect on the unproductive packing attempts), the number of customers per route, the cardinality of the routes explored etc. However, to shed light on this unexpected finding, we have performed an experimental procedure to measure some of the crucial aspects that define the computational burden of our methodology. In specific and under both models, we have applied our algorithm for both models and item rotation configurations. The following metrics were recorded:

(a) the number of times that the loading feasibility of a move had to be evaluated in terms of the loading constraints F_m,
(b) the number of times that the loading feasibility of these moves was retrieved from the SMD instances F_{smd},
(c) the number of loading feasibility examinations for complete routes F_r,
(d) the number of times that the loading feasibility of complete routes was retrieved from the route hashtable structures F_{rh},
(e) the number of loading feasibility examinations for individual arcs F_a,
(f) the number of times that the loading feasibility of individual arcs were retrieved from the arc hashtables F_{ah},
(g) the number of times that the packing heuristic was invoked F_p,
(h) the number of times that the packing heuristic was successful on finding a feasible packing F_{ps}.

Note that for the LIFO version 2L-SPD, when a route is checked in terms of the loading constraints up to two arcs have to be examined (the first route arc for the delivery items and the last route arc for the pick-up items), whereas for each arc up to three packing heuristic calls may be used (see §6.1). In addition, the arc hashtables are inapplicable.

The contents of Tables A.11 and A.12 reveal that the basic 2L-SPD model was solved more efficiently compared to the LIFO variant because the LIFO variant has considerably tighter constraints. This fact causes the following: since our local search applies the best feasible move, more tentative moves have to be evaluated regarding the loading aspects (Zachariadis and Kiranoudis, 2010). On average, the 2L-SPD model with LIFO constraints requires the evaluation of 11.3 times more moves compared to the basic 2L-SPD model, for the fixed item orientation. When item rotations are allowed, 9.9 times more moves are investigated. This implies that more routes have to be examined under the LIFO version of 2L-SPD. On average, under 2|O|SPD-L, 6.3 times more routes have to be examined compared to the 2|O|SPD case. When item rotations are allowed, the consideration of LIFO constraints requires the examination of 5.5 times more routes. This picture changes in terms of the necessary checks for the individual arcs: the LIFO version requires 1.1 and 0.9 times the number of checks required for the fixed and non-fixed item orientation version of 2L-SPD, respectively. This is because the LIFO version requires at most two arc checks per route, whereas the basic 2L-SPD model requires that all route arcs are examined. However, the filtering effect of the arc hashtables (strengthened by the fact that no LIFO constraints are imposed which causes each customer pick-up and delivery combination to be associated with only one feasibility status) has a major impact, as 79.6% (80.4%) of the arc feasibility checks are completed by accessing the corresponding arc hashtables for the fixed orientation (for the rotations configuration). As a result, the calls to the packing heuristic are considerably greater in number under the LIFO version: 9.8 times greater for 2|O|SPD-L and 7.6 times greater for 2|R|SPD-L. Furthermore, due to the tighter nature of the LIFO constraints, the packing heuristic method exhibits a rather low percentage of successful applications. On average, 54.2% and 57.5% packing heuristic runs are successful for the 2|O|SPD-L and 2|R|SPD-L versions, respectively. These success rates are almost doubled for the non-LIFO model (2|O|SPD: 94.2%, 2|R|SPD: 96.2%). This implies that for the tighter LIFO version, the overall method applies more packing attempts for identifying feasible loading structures which in practice determines the total computational time required.

8. Conclusions

The present paper introduces the Vehicle Routing Problem with Simultaneous Pick-Ups and Deliveries and Two-Dimensional Loading Constraints (2L-SPD). The main characteristic of the examined problem is that customers require both delivery and pick-up transportation service. The transported products/materials correspond to rectangular items which cannot be stacked in the vehicle loading spaces. The 2L-SPD model belongs to the integrated vehicle routing and packing problems which jointly call for the minimization of the routing costs and the determination of feasible loading patterns for the transported products. The characteristic of simultaneous pick-up and delivery service makes the loading feasibility investigation a very challenging task: along their trips, vehicles carry different item sets, so that feasible loading patterns must be identified for each of these item sets, or in other words, for each of the arcs traversed. Thus, the 2L-SPD model generalizes the Vehicle Routing Problem with Two Dimensional Loading Constraints (2L-CVRP) (Gendreau et al., 2008) which calls for the

determination of feasible loading patterns only for the fully loaded depot-leaving arcs. Except for the basic 2L-SPD version, we introduce the 2L-SPD with LIFO constraints. Contrary to the basic 2L-SPD model, under the LIFO variant, item re-arrangements along the vehicle trips are not allowed.

To solve both versions of the examined model, we have designed a new 2L-SPD solution approach. The basic algorithmic ingredients are a local search method for optimizing the routing aspects and a two-dimensional packing heuristic for generating feasible loading structures for the transported items. Both algorithmic components are extensions of our previous work on the 2L-CVRP model (Zachariadis et al., 2013). They have been integrated into a 2L-SPD solution framework which makes use of loading feasibility memory strategies specifically designed for the 2L-SPD, in order to drastically reduce the required computational effort.

To test the effectiveness of our solution approach, we have conducted a series of runs on the 2L-CVRP and VRPSPD models. Fine quality results are obtained, improving and matching several best known solution scores. In addition, our algorithm was applied to a series of newly constructed 2L-SPD benchmark instances. The obtained results indicate that our method is stable and capable of achieving very high utilization of the vehicle loading spaces. Moreover, the computational times required by our algorithm can be deemed reasonable, taking into account both the great complexity of the examined problem and the large scale of the benchmark problems used in our experiments.

In terms of further work in the area of composite routing and packing problems, we point out two interesting research directions. The first one is the introduction and examination of routing problems integrated with loading constraints that apart from minimizing the routing costs, will call for the maximization of the transported items. In other words, the loading requirements will not be regarded as a hard constraint of the underlying model, but rather as a soft constraint aimed at maximizing the items loaded onto the vehicle spaces, just like in the case of knapsack problems. Another interesting research direction is the detailed examination of the cargo handling (loading/unloading/rearrangement) activities that must take place at each service location. Models which will associate the packing arrangements of items with the required time for such activities should be developed. Then, a maximum time for handling purposes could be integrated into the underlying model. In addition, this aspect could be incorporated into the objective function which could jointly call for the minimization of the total travel time and the required cargo handling time.

Acknowledgments

The work presented in this paper was supported by the Program for supporting Post-Doctoral Research (Siemens) 2013-14 of the Greek State Scholarship Foundation (IKY).

The authors would like to thank the referees and the editor for their valuable remarks that greatly improved the manuscript.

References

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

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

Crainic T G, Perboli G, Tadei R (2008). Extreme point-based heuristics for three- dimensional bin packing. *INFORMS Journal on Computing* **20**: 368–384.

Dell'Amico M, Righini G, Salani M (2006). A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science* **40**: 235–247.

Dominguez O, Juan A, Faulin J (2014). A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations. *International Transactions in Operational Research* **21**: 375–398.

Duhamel C, Lacomme P, Quilliot A, Toussaint H (2011). A multi-start evolutionary local search for the two dimensional loading capacitated vehicle routing problem. *Computers & Operations Research* **38**: 617–640.

Fuellerer G, Doerner K F, Hartl R F, Iori M (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. Computers & Operations Research **36:** 655–673.

Fuellerer G, Doerner K, Hartl R, Iori M (2010). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research* **201**: 751–759.

Gendreau M, Iori M, Laporte G, Martello S (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science* **40**: 342–350.

Gendreau M, Iori M, Laporte G, Martello S (2008). A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks* **51**: 514–518.

Iori M, Martello S (2010). Routing problems with loading constraints. TOP **18**: 4–27.

Iori M, Martello S (2013). An annotated bibliography of combined routing and loading problems. Yugoslav Journal of Operations Research **23**: 311-326.

Iori M, Salazar-Gonzãlez J-J, Vigo D (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science* **41**: 253–264.

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

Leung SCH, Zhou X, Zhang D, Zheng J (2011). Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem. *Computers & Operations Research* **38**: 205–215.

Lodi A, Martello S, Monaci M (2002). Two-dimensional packing problems: A survey. *European Journal of Operational Research* **141**: 241-252.

Lodi A, Martello S, Vigo D (1999). Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing* **11**: 345–357.

Malapert A, Guéret C, Jussien N, Langevin A, Rousseau L-M (2008). Two-dimensional Pickup and Delivery Routing Problem with Loading Constraints. 1st Workshop on Bin Packing and Placement Constraints (BPPC'08), Paris.

Männel D, Bortfeldt A (2013). The pickup and delivery problem with three-dimensional loading constraints: formulation, solution approach and preliminary computational results. 10th ESICUP Meeting, Lille, France.

Martello, S, Pisinger D, Vigo D (2000). The three-dimensional bin packing problem. *Operations Research* **48**: 256-267.

Perboli G, Gobbato L, Perfetti F (2014). Packing Problems in Transportation and Supply Chain: New Problems and Trends. *Procedia - Social and Behavioral Sciences* **111**: 672-681.

Ruan Q, Zhang Z, Miao L, Shen H (2011). A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research* **40**: 1579-1589.

Strodl J, Doerner K, Tricoire F, Hartl R (2010). On Index Structures in Hybrid Metaheuristics for Routing Problems with Hard Feasibility Checks: An Application to the 2-Dimensional Loading. In: Blesa, M., Blum, C., Raidl, G.,Roli, A., Sampels, M. (Eds.), Hybrid Metaheuristics, Lecture Notes in Computer Science, vol. 6373. Springer, Berlin/Heidelberg, pp. 160–173.

Subramanian A, Uchoa E, Pessoa A A, Ochi L S (2013a). Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optimization Letters* **7**: 1569-1581.

Subramanian A, Uchoa E, Pessoa A A, Ochi L S (2013b). A hybrid algorithm for a class of vehicle routing problems, Computers & Operations Research 40, 2519-2531.

Tang Montanè F A, Galvão R D (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers and Operations Research* **33**: 595–619.

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

Wei L, Zhang Z, Zhang D, Lim A (2015). A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research* **243**: 798–814.

Whittley I M, Smith G D (2004). The attribute based hill climber. *Journal of Mathematical Modelling and Algorithms* **3**: 167–178.

Zachariadis E E, Tarantilis C D, Kiranoudis C T (2013). Integrated distribution and loading planning via a compact metaheuristic algorithm. *European Journal of Operational Research* **228**: 56-71.

Zachariadis E E, Tarantilis C D, Kiranoudis C T (2009). A guided tabu search for the vehicle routing problem with two dimensional loading constraints. *European Journal Operations Research* **195**: 729–743.

Zachariadis E E, Kiranoudis C T (2010). A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. *Computers & Operations Research* **37**: 2089–2105.

Zhu W, Qin H, Lim A, Wang L, (2012). A two-stage tabu search algorithm with enhanced packing heuristics for the 3L-CVRP and M3L-CVRP. *Computers & Operations Research* **39**: 2178–2195.

Appendix A

Table A.1. The 2L-SPD benchmark instances

		Class 2					Class 3					Class 4					Class 5				
Inst	**n**	**d**	**p**	**k**	**%d**	**%p**	**d**	**p**	**k**	**%d**	**%p**	**d**	**p**	**k**	**%d**	**%p**	**d**	**p**	**k**	**%d**	**%p**
1	15	13	11	2	58	60	19	12	2	78	44	19	18	2	78	62	18	27	2	45	76
2	15	11	14	3	38	49	14	17	3	44	54	23	17	2	74	59	31	17	2	59	39
3	20	13	16	3	56	57	25	21	3	73	55	20	24	2	75	80	29	20	2	82	53
4	20	20	12	4	56	30	26	17	3	67	49	25	25	4	46	49	29	33	2	67	73
5	21	10	21	3	32	65	18	19	2	68	83	30	11	3	79	22	25	32	2	53	79
6	21	12	21	3	42	66	19	21	3	58	68	26	31	4	46	62	27	29	2	60	79
7	22	18	14	3	69	49	20	21	2	80	86	28	23	3	62	50	36	19	3	66	31
8	22	21	8	3	78	27	24	18	3	64	54	21	27	3	55	59	24	28	2	56	57
9	25	23	17	4	67	47	34	27	3	89	74	35	28	4	69	51	46	45	3	76	66
10	29	22	21	3	72	75	23	26	3	60	72	27	45	4	48	79	36	50	3	62	85
11	29	27	16	4	78	37	34	28	4	70	60	28	46	4	62	82	48	43	3	78	70
12	30	26	24	5	56	56	27	29	3	78	76	37	45	5	55	63	55	46	4	71	60
13	32	19	25	4	49	72	24	32	3	70	88	39	39	3	88	92	58	44	3	89	68
14	32	25	22	3	81	71	32	25	3	82	71	24	41	3	63	79	46	41	3	69	62
15	32	20	28	3	64	87	33	26	3	93	74	45	39	4	77	67	64	50	4	80	63
16	35	27	29	6	43	59	37	37	5	64	61	46	47	5	67	73	53	61	4	58	76
17	40	35	25	5	76	53	36	37	4	71	75	51	45	5	75	69	64	63	4	71	69
18	44	29	37	4	71	90	39	48	5	65	85	47	65	5	67	88	52	70	4	61	85
19	50	41	41	5	86	84	52	51	6	77	74	73	61	6	88	70	82	75	4	91	91
20	71	55	49	8	80	67	70	81	8	71	85	90	88	8	80	82	110	116	7	83	74
21	75	50	64	8	66	81	86	78	9	79	75	82	86	8	68	81	96	106	6	83	88
22	75	51	61	8	68	86	85	69	8	88	75	109	89	9	84	71	113	123	7	78	83
23	75	49	63	8	71	79	74	81	8	79	86	95	84	8	90	77	110	115	7	78	86
24	75	68	56	10	77	61	70	82	8	75	88	110	85	9	87	67	100	115	8	63	77
25	100	77	80	11	78	80	101	111	11	80	82	132	122	11	84	81	163	148	10	75	69
26	100	72	75	10	78	81	95	103	10	77	88	129	118	11	84	75	154	156	9	82	84
27	100	73	79	10	77	83	108	103	11	85	79	112	133	10	79	92	156	164	9	82	92
28	120	104	79	12	90	69	116	126	13	74	86	149	150	12	87	87	181	203	10	87	94
29	134	100	97	13	79	78	143	119	14	83	70	157	185	14	78	93	193	229	12	78	96
30	150	111	114	15	79	82	159	139	16	86	77	191	175	16	86	76	210	223	12	87	88
31	199	170	137	20	88	72	192	210	21	76	86	251	262	21	82	90	316	286	17	91	81
32	199	148	151	20	77	84	197	207	20	79	86	248	249	20	83	85	308	281	16	93	86
33	199	155	146	20	83	75	194	213	21	79	85	250	249	21	86	83	305	272	17	94	78
34	240	185	185	24	83	81	222	268	25	77	91	303	301	26	83	82	372	348	21	91	81
35	252	174	193	24	75	85	248	259	25	84	86	311	323	25	88	91	395	367	20	94	91
36	255	182	205	24	80	89	243	268	25	84	91	313	293	25	90	78	432	354	22	94	78

n: number of customers, **d**: total delivery items, **p**: total pick-up items, **k**: number of available vehicles, **%d**: total area of delivery items divided by the total loading area of the vehicles, **%p**: total area of pick-up items divided by the total loading area of the vehicles

Table A.2. The 2L-SPD benchmark instances used for the LIFO version of the 2L-SPD model

		Class 4					Class 5				
Inst	**n**	**d**	**p**	**k**	**%d**	**%p**	**d**	**p**	**k**	**%d**	**%p**
L1	15	13	17	3	0.41	0.45	15	23	3	0.39	0.53
L 2	15	18	13	3	0.62	0.44	29	16	3	0.53	0.39
L3	20	14	21	4	0.37	0.56	24	17	3	0.64	0.53
L4	20	20	18	4	0.46	0.44	25	30	4	0.53	0.55
L5	21	26	9	4	0.69	0.11	22	28	3	0.43	0.60
L6	21	22	31	4	0.38	0.59	26	25	4	0.45	0.49
L7	22	23	21	3	0.46	0.39	32	19	3	0.56	0.28
L8	22	20	24	3	0.51	0.51	20	27	3	0.50	0.54
L9	25	29	30	5	0.44	0.42	41	41	4	0.64	0.61
L10	29	22	42	4	0.38	0.69	33	44	4	0.40	0.51
L11	29	22	40	5	0.37	0.52	43	41	4	0.48	0.49
L12	30	29	37	4	0.47	0.57	47	44	4	0.54	0.55
L13	32	32	35	4	0.51	0.59	55	40	5	0.54	0.38
L14	32	19	39	3	0.46	0.67	44	38	4	0.47	0.40
L15	32	39	37	5	0.53	0.48	61	45	5	0.60	0.42
L16	35	36	38	5	0.49	0.52	51	56	5	0.53	0.61
L17	40	38	42	5	0.53	0.60	61	58	5	0.71	0.53
L18	44	42	57	6	0.48	0.59	49	67	6	0.37	0.52
L19	50	58	52	7	0.57	0.49	77	71	6	0.54	0.55
L20	71	81	79	11	0.50	0.53	100	106	9	0.56	0.48
L21	75	76	73	9	0.54	0.58	90	101	8	0.55	0.59
L22	75	91	73	10	0.61	0.50	106	117	10	0.49	0.53
L23	75	80	73	11	0.53	0.46	97	101	9	0.50	0.53
L24	75	84	78	10	0.55	0.51	91	101	9	0.48	0.55
L25	100	108	106	14	0.52	0.52	151	139	12	0.54	0.51
L26	100	111	105	14	0.55	0.50	144	141	11	0.60	0.57

n: number of customers, **d**: total delivery items, **p**: total pick-up items, **k**: number of available vehicles, **%d**: total area of delivery items divided by the total loading area of the vehicles, **%p**: total area of pick-up items divided by the total loading area of the vehicles

Table A.3. Summary of the results obtained for the 2|UO|L version of 2L-CVRP

Inst	Class 2				Class 3				Class 4				Class 5			
	bst	**avg**	**t**	**%g**	**bst**	**avg**	**t**	**%g**	**bst**	**avg**	**t**	**%g**	**bst**	**avg**	**t**	**%g**
1	278.73	278.73	2.5	0.00	284.52	284.52	2.4	0.00	282.95	282.95	1.7	0.00	278.73	278.73	1.4	0.00
2	334.96	334.96	0.2	0.00	352.16	352.16	2.5	0.00	334.96	334.96	0.3	0.00	334.96	334.96	1.6	0.00
3	387.70	387.70	1.2	0.00	394.72	394.72	0.1	0.00	364.45	364.45	1.1	0.00	358.40	358.40	1.8	0.00
4	430.88	430.88	0.1	0.00	430.88	430.88	0.9	0.00	447.37	447.37	0.8	0.00	430.88	430.88	1.5	0.00
5	375.28	375.28	0.3	0.00	381.69	381.69	0.7	0.00	383.87	383.87	0.2	0.00	375.28	375.28	2.6	0.00
6	495.85	495.85	0.9	0.00	498.16	498.16	0.1	0.00	498.32	498.32	2.4	0.00	495.85	495.85	1.1	0.00
7	725.46	725.46	2.0	0.00	678.75	678.75	0.1	0.00	700.72	700.72	2.0	0.00	657.77	657.77	1.9	0.00
8	674.55	674.55	2.1	0.00	738.43	738.43	1.9	0.00	692.47	692.47	1.0	0.00	609.90	609.90	1.3	0.00
9	607.65	607.65	1.6	0.00	607.65	607.65	0.9	0.00	625.10	625.10	0.4	0.00	607.65	607.65	1.2	0.00
10	689.68	689.68	6.7	0.00	620.33	620.33	7.9	0.00	710.87	710.87	1.6	0.00	686.03	686.03	10.8	0.00
11	694.47	703.43	10.3	1.29	706.73	706.73	9.8	0.00	786.85	786.85	1.9	0.00	624.82	624.82	15.4	0.00
12	610.57	610.57	1.1	0.00	610.00	610.00	1.1	0.00	614.23	614.23	1.2	0.00	610.23	610.23	1.4	0.00
13	2585.72	2585.93	13.8	0.01	2436.56	2436.56	17.2	0.00	2607.66	2607.66	21.6	0.00	2334.78	2334.78	20.6	0.00
14	1038.20	1038.21	75.2	0.00	1003.52	1003.52	81.0	0.00	981.00	981.00	119.2	0.00	876.33	879.43	107.2	0.35
15	1017.95	1018.13	9.0	0.02	1156.00	1163.44	10.3	0.64	1187.30	1193.95	11.1	0.56	1160.20	1160.20	13.7	0.00
16	698.61	698.61	1.7	0.00	698.61	698.61	2.2	0.00	703.35	703.35	2.6	0.00	698.61	698.61	2.6	0.00
17	870.86	871.47	4.9	0.07	861.79	861.79	5.5	0.00	861.79	861.79	7.4	0.00	861.79	861.80	8.1	0.00
18	1004.99	1009.98	11.7	0.50	1081.44	1081.44	14.9	0.00	1119.55	1123.42	14.0	0.35	917.93	918.41	17.5	0.05
19	759.30	760.01	21.7	0.09	771.66	772.61	29.1	0.12	779.72	781.24	31.1	0.19	651.97	651.99	33.4	0.00
20	517.06	522.61	220.1	1.07	521.41	521.81	247.2	0.08	540.34	540.57	247.6	0.04	472.54	474.02	295.2	0.31
21	997.63	999.40	64.8	0.18	1118.98	1123.84	68.4	0.43	974.08	982.65	76.3	0.88	878.63	884.29	85.4	0.64
22	1035.81	1040.83	62.2	0.48	1050.42	1057.50	65.9	0.67	1052.48	1052.64	92.2	0.02	933.60	938.78	85.6	0.56
23	1035.18	1040.93	79.6	0.56	1078.68	1080.53	80.7	0.17	1075.36	1078.35	82.1	0.28	936.53	939.61	91.2	0.33
24	1178.07	1182.37	144.4	0.37	1080.88	1084.11	134.9	0.30	1108.87	1116.16	194.3	0.66	1044.56	1048.59	178.4	0.39
25	1406.98	1410.19	285.7	0.23	1365.37	1377.39	322.2	0.88	1404.47	1411.15	370.6	0.48	1160.48	1165.85	453.6	0.46
26	1279.48	1285.19	169.5	0.45	1344.08	1345.30	229.9	0.09	1400.38	1401.99	240.9	0.12	1225.29	1227.77	252.7	0.20
27	1316.93	1326.44	556.9	0.72	1374.10	1378.32	668.9	0.31	1319.35	1323.58	615.4	0.32	1248.94	1254.11	704.0	0.41
28	2552.53	2585.88	1198.4	1.31	2598.77	2638.26	1414.5	1.52	2604.59	2641.63	1452.3	1.42	2316.66	2329.18	1422.9	0.54
29	2204.33	2235.10	771.6	1.40	2092.42	2137.68	873.0	2.16	2251.26	2270.21	936.6	0.84	2132.36	2152.03	968.5	0.92
30	1806.38	1825.32	588.8	1.05	1834.64	1845.90	658.4	0.61	1830.77	1847.69	799.3	0.92	1535.32	1541.74	938.7	0.42
31	2264.34	2290.07	828.3	1.14	2276.29	2300.17	928.6	1.05	2382.41	2404.81	915.8	0.94	2005.56	2019.89	1069.1	0.71
32	2261.24	2291.50	1244.6	1.34	2237.73	2260.72	1340.1	1.03	2258.25	2284.36	1675.5	1.16	1966.18	1980.13	1530.1	0.71
33	2253.45	2286.72	854.8	1.48	2360.92	2381.23	815.2	0.86	2386.72	2411.32	908.2	1.03	1988.95	2005.05	1009.9	0.81
34	1178.40	1187.73	1565.6	0.79	1204.52	1213.23	1813.9	0.72	1206.03	1215.39	2475.6	0.78	1022.68	1032.02	2201.5	0.91
35	1373.42	1383.47	1200.7	0.73	1445.42	1456.46	1250.4	0.76	1498.02	1508.50	1619.8	0.70	1248.55	1253.88	1489.7	0.43
36	1695.37	1711.66	1398.6	0.96	1772.81	1784.95	1510.9	0.68	1652.99	1663.32	1722.3	0.63	1484.18	1490.91	1732.3	0.45
avg			*316.7*	*0.45*			*350.3*	*0.36*			*406.8*	*0.34*			*409.8*	*0.27*

bst: the best solution score obtained over the ten runs, avg: the average solution score over the ten runs, **t**: the average computational time elapsed for obtaining the final solution over the ten runs (in CPU sec), **%g:** the percent deviation between our best and average scores (= 100(avg-bst)/bst)

Table A.4. Comparison of the best known solution scores for the 2|UO|L version of 2L-CVRP

Inst	Class 2			Class 3			Class 4			Class 5		
	bst	**BKS**	**%g**	**bst**	**BKS**	**%g**	**bst**	**BKS**	**%g**	**bst**	**BKS**	**%g**
1	**278.73**	**278.73**	0.00	**284.52**	**284.52***	0.00	**282.95**	**282.95**	0.00	**278.73**	**278.73**	0.00
2	**334.96**	**334.96**	0.00	**352.16**	**352.16**	0.00	**334.96**	**334.96**	0.00	**334.96**	**334.96**	0.00
3	**387.70**	**387.70**	0.00	**394.72**	**394.72**	0.00	**364.45**	**364.45**	0.00	**358.40**	**358.40**	0.00
4	**430.88**	**430.88**	0.00	**430.88**	**430.88**	0.00	**447.37**	**447.37**	0.00	**430.88**	**430.88**	0.00
5	**375.28**	**375.28**	0.00	**381.69**	**381.69**	0.00	**383.87**	**383.87**	0.00	**375.28**	**375.28**	0.00
6	**495.85**	**495.85**	0.00	498.16	**497.17**	0.20	**498.32**	**498.32**	0.00	495.85	**495.75**	0.02
7	**725.46**	**725.46**	0.00	**678.75**	**678.75**	0.00	**700.72**	**700.72**	0.00	**657.77**	**657.77**	0.00
8	**674.55**	**674.55**	0.00	**738.43**	**738.43**	0.00	**692.47**	**692.47**	0.00	**609.90**	**609.90**	0.00
9	**607.65**	**607.65**	0.00	**607.65**	**607.65**	0.00	625.10	**621.23**	0.62	**607.65**	**607.65**	0.00
10	**689.68**	**689.68**	0.00	620.33	**615.68**	0.76	**710.87**	**710.87**	0.00	686.03	**678.66**	1.09
11	694.47	**693.45**	0.15	**706.73**	**706.73**	0.00	786.85	**784.88**	0.25	**624.82**	**624.82**	0.00
12	**610.57**	**610.57**	0.00	**610.00**	**610.00**	0.00	**614.23**	**614.23**	0.00	610.23	**610.00**	0.04
13	**2585.72**	**2585.72**	0.00	**2436.56**	**2436.56**	0.00	2607.66	**2548.06**	2.34	**2334.78**	**2334.78**	0.00
14	1038.20	**1038.09**	0.01	1003.52	**994.61**	0.90	**981.00**	**981.00**	0.00	876.33	**874.55**	0.20
15	1017.95	**1013.29**	0.46	1156.00	**1154.66**	0.12	1187.30	**1181.30**	0.51	1160.20	**1159.94**	0.02
16	**698.61**	**698.61**	0.00	**698.61**	**698.61**	0.00	**703.35**	**703.35**	0.00	**698.61**	**698.61**	0.00
17	870.86	**863.66**	0.83	**861.79**	**861.79**	0.00	**861.79**	**861.79**	0.00	**861.79**	**861.79**	0.00
18	**1004.99**	**1004.99**	0.00	1081.44	**1069.45**	1.12	1119.55	**1118.57**	0.09	**917.93**	917.94	0.00
19	759.30	**754.53**	0.63	**771.66**	771.74	-0.01	779.72	**775.87**	0.50	651.97	**644.59**	1.14
20	**517.06**	525.75	-1.65	521.41	**521.31**	0.02	540.34	**538.86**	0.27	472.54	**471.64**	0.19
21	997.63	**992.83**	0.48	1118.98	**1118.11**	0.08	974.08	**970.90**	0.33	878.63	**877.75**	0.10
22	1035.81	**1035.66**	0.01	**1050.42**	1052.98	-0.24	1052.48	**1045.91**	0.63	933.60	**932.11**	0.16
23	**1035.18**	**1035.18**	0.00	**1078.68**	1079.03	-0.03	1075.36	**1071.30**	0.38	936.53	**935.33**	0.13
24	**1178.07**	**1178.07**	0.00	**1080.88**	**1080.88**	0.00	1108.87	**1108.34**	0.05	1044.56	**1028.04**	1.61
25	**1406.98**	1409.24	-0.16	**1365.37**	1369.26	-0.28	**1404.47**	1404.73	-0.02	1160.48	**1150.69**	0.85
26	1279.48	**1272.87**	0.52	**1344.08**	1344.10	0.00	1400.38	**1394.19**	0.44	1225.29	**1213.88**	0.94
27	1316.93	**1312.68**	0.32	1374.10	**1370.40**	0.27	1319.35	**1316.19**	0.24	1248.94	**1244.81**	0.33
28	2552.53	**2551.18**	0.05	**2598.77**	2601.73	-0.11	2604.59	**2576.44**	1.09	2316.66	**2305.58**	0.48
29	2204.33	**2199.79**	0.21	2092.42	**2085.70**	0.32	2251.26	**2242.74**	0.38	2132.36	**2128.37**	0.19
30	**1806.38**	1807.64	-0.07	1834.64	**1829.72**	0.27	1830.77	**1826.10**	0.26	1535.32	**1520.54**	0.97
31	2264.34	**2252.58**	0.52	2276.29	**2276.14**	0.01	2382.41	**2378.08**	0.18	2005.56	**1986.23**	0.97
32	2261.24	**2253.79**	0.33	2237.73	**2235.57**	0.10	**2258.25**	2259.83	-0.07	1966.18	**1954.15**	0.62
33	**2253.45**	2259.29	-0.26	2360.92	**2348.25**	0.54	2386.72	**2372.18**	0.61	1988.95	**1971.40**	0.89
34	1178.40	**1176.03**	0.20	1204.52	**1201.42**	0.26	1206.03	**1191.43**	1.23	1022.68	**1016.88**	0.57
35	1373.42	**1372.70**	0.05	1445.42	**1444.88**	0.04	1498.02	**1492.98**	0.34	1248.55	**1237.51**	0.89
36	1695.37	**1693.83**	0.09	1772.81	**1771.73**	0.06	1652.99	**1650.01**	0.18	1484.18	**1475.24**	0.61
avg			*0.08*			*0.12*			*0.30*			*0.36*

bst: our best solution score, **BKS**: the best known solution score, **g%**: the percent deviation between our best and the BKS scores (= 100(bst-BKS)/BKS).
Sources for the BKS scores: Wei et al. (2015), Dominguez et al. (2014), Zachariadis et al. (2013), Leung et al. (2011) and Duhamel et al. (2011).
Note that some BKS values have been obtained during the development of the VNS method (Wei et al, 2015) and were communicated to us directly by the authors.
*Lower scores for these instances were mistakenly reported in Dominguez et al. (2014). This was clarified by contacting the authors of Dominguez et al. (2014).

Table A.5. Summary of the results obtained for the 2|UR|L version of 2L-CVRP

Inst	Class 2				Class 3				Class 4				Class 5			
	bst	avg	t	%g	bst	avg	t	%g	bst	avg	t	%g	bst	avg	t	%g
1	278.73	278.73	0.6	0.00	284.11	284.10	0.7	0.00	282.95	282.95	0.6	0.00	278.73	278.73	1.5	0.00
2	334.96	334.96	0.2	0.00	352.16	352.16	1.7	0.00	334.96	334.96	0.4	0.00	334.96	334.96	0.4	0.00
3	385.29	385.29	1.5	0.00	385.32	385.32	1.2	0.00	362.41	362.41	2.5	0.00	358.40	358.40	2.2	0.00
4	430.89	430.88	1.1	0.00	430.89	430.88	0.8	0.00	447.37	447.37	0.5	0.00	430.89	430.88	0.3	0.00
5	375.28	375.28	1.5	0.00	379.94	379.94	1.1	0.00	383.88	383.87	1.9	0.00	375.28	375.28	2.1	0.00
6	495.85	495.85	0.7	0.00	498.16	498.16	0.8	0.00	498.32	498.32	2.9	0.00	495.85	495.85	0.8	0.00
7	715.02	715.02	1.3	0.00	664.96	664.96	0.8	0.00	686.26	686.26	2.3	0.00	657.77	657.77	1.7	0.00
8	665.17	665.17	1.5	0.00	738.43	738.43	1.8	0.00	688.32	692.47	1.0	0.60	609.90	609.90	2.0	0.00
9	607.65	607.65	1.5	0.00	607.65	607.65	0.4	0.00	625.10	625.10	1.6	0.00	607.65	607.65	3.6	0.00
10	667.42	667.42	0.8	0.00	591.16	591.16	0.2	0.00	703.64	703.64	2.2	0.00	678.66	680.26	10.7	0.24
11	666.16	666.16	1.9	0.00	699.35	699.35	0.2	0.00	773.58	773.58	1.5	0.00	624.82	624.82	2.6	0.00
12	610.00	610.00	4.4	0.00	610.00	610.00	4.7	0.00	610.23	614.23	2.3	0.66	610.00	610.00	6.3	0.00
13	2502.65	2502.65	2.2	0.00	2377.39	2377.39	5.9	0.00	2500.85	2500.85	1.3	0.00	2334.59	2334.59	7.0	0.00
14	1029.34	1029.34	16.6	0.00	988.80	988.79	36.3	0.00	968.21	972.89	110.8	0.48	871.22	871.22	388.8	0.00
15	1001.51	1001.51	32.2	0.00	1116.07	1120.82	407.8	0.43	1164.63	1164.63	231.4	0.00	1160.20	1160.20	11.4	0.00
16	698.61	698.61	1.0	0.00	698.61	698.61	0.6	0.00	703.35	703.35	1.5	0.00	698.61	698.61	0.6	0.00
17	861.79	861.79	11.6	0.00	861.79	861.79	9.8	0.00	861.79	861.83	16.8	0.01	861.79	861.79	12.2	0.00
18	982.44	982.63	64.2	0.02	1009.62	1009.62	19.6	0.00	1100.52	1100.52	14.1	0.00	917.94	918.90	70.1	0.11
19	711.97	718.20	30.6	0.87	751.56	753.21	110.4	0.22	755.04	760.37	133.2	0.71	644.59	644.59	8.5	0.00
20	488.69	488.73	139.2	0.01	511.46	511.50	119.0	0.01	535.03	535.03	87.5	0.00	468.60	469.19	210.8	0.13
21	962.10	963.62	182.3	0.16	1087.79	1091.39	204.6	0.33	958.58	963.66	262.0	0.53	870.82	875.36	459.4	0.52
22	993.50	996.75	454.1	0.33	1028.33	1031.86	312.9	0.34	1042.01	1042.61	233.6	0.06	930.83	932.73	288.1	0.20
23	991.99	994.09	217.0	0.21	1044.06	1049.79	233.0	0.55	1048.43	1049.82	237.1	0.13	926.68	929.10	205.2	0.26
24	1142.02	1147.73	102.9	0.50	1064.38	1069.64	121.8	0.49	1086.09	1093.28	166.7	0.66	1042.37	1045.45	76.0	0.30
25	1348.66	1354.45	260.7	0.43	1325.24	1334.69	336.9	0.71	1374.79	1381.14	457.2	0.46	1150.04	1155.47	624.5	0.47
26	1255.16	1260.87	132.2	0.45	1312.94	1326.65	159.2	1.04	1378.21	1395.66	255.2	1.27	1213.03	1215.88	181.0	0.23
27	1266.89	1278.20	316.4	0.89	1332.15	1345.11	280.2	0.97	1289.02	1298.63	436.9	0.75	1237.05	1247.30	447.3	0.83
28	2482.76	2498.04	616.0	0.62	2544.39	2571.22	570.0	1.05	2514.87	2542.76	835.9	1.11	2287.98	2309.40	705.2	0.94
29	2128.53	2153.92	1280.3	1.19	2051.52	2083.08	1364.3	1.54	2209.58	2225.09	1404.6	0.70	2125.35	2174.76	1463.5	2.32
30	1744.11	1759.50	1206.5	0.88	1772.71	1789.80	527.4	0.96	1795.94	1803.67	1074.9	0.43	1517.86	1526.08	852.2	0.54
31	2154.33	2170.44	796.7	0.75	2196.40	2212.07	900.1	0.71	2326.63	2344.61	1177.6	0.77	1980.17	1996.76	1270.8	0.84
32	2169.66	2190.96	692.9	0.98	2179.06	2201.05	879.2	1.01	2212.94	2230.18	1336.7	0.78	1949.14	1966.57	1213.3	0.89
33	2161.03	2181.62	958.7	0.95	2284.46	2308.71	877.5	1.06	2326.73	2349.92	1548.4	1.00	1968.32	1984.93	1545.1	0.84
34	1120.44	1130.99	1112.2	0.94	1163.05	1173.65	1043.1	0.91	1171.04	1182.12	1159.7	0.95	1014.20	1022.20	1322.2	0.79
35	1312.88	1324.86	1355.2	0.91	1397.77	1407.12	1272.9	0.67	1460.85	1471.03	1729.2	0.70	1227.13	1238.13	1826.1	0.90
36	1623.54	1635.84	2493.8	0.76	1706.70	1721.36	3303.0	0.86	1604.55	1620.20	3969.1	0.98	1462.33	1472.73	3831.7	0.71
avg			*347.0*	*0.33*			*364.2*	*0.39*			*469.5*	*0.38*			*473.7*	*0.34*

The notation of Table 3 is used

Table A.6. Comparison of the best known solution scores for the 2|UR|L version of 2L-CVRP

Inst	Class 2			Class 3			Class 4			Class 5		
	bst	**BKS**	**%g**	**bst**	**BKS**	**%g**	**bst**	**BKS**	**%g**	**bst**	**BKS**	**%g**
1	**278.73**	**278.73**	0.00	**284.10**	**284.10***	0.41	**282.95**	**282.95**	0.00	**278.73**	**278.73**	0.00
2	**334.96**	**334.96**	0.00	**352.16**	**352.16**	0.00	**334.96**	**334.96**	0.00	**334.96**	**334.96**	0.00
3	385.29	**380.35**	1.30	**385.32**	**385.32**	0.00	362.41	**358.40**	1.12	**358.40**	**358.40**	0.00
4	**430.89**	430.88	0.00	**430.89**	**430.88**	0.00	**447.37**	**447.37**	0.00	430.89	**430.88**	0.00
5	**375.28**	**375.28**	0.00	**379.94**	**379.94**	0.00	383.88	**383.87**	0.00	**375.28**	**375.28**	0.00
6	**495.85**	**495.85**	0.00	**498.16**	**498.16**	0.00	**498.32**	**498.32**	0.00	**495.85**	**495.85**	0.00
7	**715.02**	**715.02**	0.00	**664.96**	**664.96**	0.00	**686.26**	**686.26**	0.00	**657.77**	**657.77**	0.00
8	**665.17**	**665.17**	0.00	**738.43**	**738.43**	0.00	**688.32**	**688.32**	0.00	**609.90**	**609.90**	0.00
9	**607.65**	**607.65**	0.00	**607.65**	**607.65**	0.00	**625.10**	**625.10**	0.00	**607.65**	**607.65**	0.00
10	**667.42**	**667.42**	0.00	**591.16**	615.36	-3.93	**703.64**	**703.63**	0.00	**678.66**	680.26	-0.24
11	666.16	**664.48**	0.25	**699.35**	**699.35**	0.00	**773.58**	**773.58**	0.00	**624.82**	**624.82**	0.00
12	**610.00**	**610.00**	0.00	**610.00**	**610.00**	0.00	**610.23**	614.23	-0.65	**610.00**	**610.00**	0.00
13	**2502.65**	**2502.65**	0.00	**2377.39**	**2377.39**	0.00	**2500.85**	2533.79	-1.30	**2334.59**	2334.78	-0.01
14	**1029.34**	**1029.34**	0.00	988.80	**988.79**	0.00	**968.21**	981.00	-1.30	**871.22**	875.07	-0.44
15	**1001.51**	**1001.51**	0.00	**1116.07**	1120.75	-0.42	**1164.63**	1164.77	-0.01	**1160.20**	**1160.20**	0.00
16	**698.61**	**698.61**	0.00	**698.61**	**698.61**	0.00	**703.35**	**703.35**	0.00	**698.61**	**698.61**	0.00
17	**861.79**	**861.79**	0.00	**861.79**	**861.79**	0.00	**861.79**	**861.79**	0.00	**861.79**	**861.79**	0.00
18	**982.44**	988.61	-0.62	1009.62	**986.30**	2.36	**1100.52**	1100.66	-0.01	**917.94**	921.29	-0.36
19	**711.97**	726.51	-2.00	**751.56**	752.06	-0.07	**755.04**	765.51	-1.37	**644.59**	**644.59**	0.00
20	**488.69**	489.23	-0.11	**511.46**	**511.46**	0.00	535.03	**534.14**	0.17	**468.60**	472.77	-0.88
21	**962.10**	964.49	-0.25	**1087.79**	1089.75	-0.18	**958.58**	967.85	-0.96	**870.82**	886.04	-1.72
22	993.50	**976.70**	1.72	**1028.33**	1031.79	-0.34	**1042.01**	1052.60	-1.01	**930.83**	942.06	-1.19
23	991.99	**985.18**	0.69	**1044.06**	1056.56	-1.18	**1048.43**	1064.76	-1.53	**926.68**	938.25	-1.23
24	**1142.02**	1152.35	-0.90	**1064.38**	1073.01	-0.80	**1086.09**	1099.40	-1.21	**1042.37**	1046.84	-0.43
25	**1348.66**	1356.24	-0.56	**1325.24**	1353.90	-2.12	**1374.79**	1402.08	-1.95	**1150.04**	1168.87	-1.61
26	**1255.16**	1262.43	-0.58	**1312.94**	1335.80	-1.71	**1378.21**	1391.02	-0.92	**1213.03**	1220.83	-0.64
27	**1266.89**	1283.66	-1.31	**1332.15**	1354.76	-1.67	**1289.02**	1318.45	-2.23	**1237.05**	1258.12	-1.67
28	**2482.76**	2517.25	-1.37	**2544.39**	2587.25	-1.66	**2514.87**	2638.07	-4.67	**2287.98**	2322.37	-1.48
29	**2128.53**	2151.68	-1.08	**2051.52**	2067.69	-0.78	**2209.58**	2267.37	-2.55	**2125.35**	2152.26	-1.25
30	**1744.11**	1755.89	-0.67	**1772.71**	1811.22	-2.13	**1795.94**	1834.68	-2.11	**1517.86**	1542.14	-1.57
31	**2154.33**	2171.60	-0.80	**2196.40**	2246.54	-2.23	**2326.63**	2385.63	-2.47	**1980.17**	2011.88	-1.58
32	**2169.66**	2191.58	-1.00	**2179.06**	2219.26	-1.81	**2212.94**	2267.57	-2.41	**1949.14**	1983.34	-1.72
33	**2161.03**	2175.85	-0.68	**2284.46**	2325.36	-1.76	**2326.73**	2387.22	-2.53	**1968.32**	2001.26	-1.65
34	**1120.44**	1140.83	-1.79	**1163.05**	1176.71	-1.16	**1171.04**	1208.19	-3.07	**1014.20**	1036.16	-2.12
35	**1312.88**	1340.41	-2.05	**1397.77**	1437.30	-2.75	**1460.85**	1503.42	-2.83	**1227.13**	1256.34	-2.33
36	**1623.54**	1679.27	-3.32	**1706.70**	1739.36	-1.88	**1604.55**	1670.84	-3.97	**1462.33**	1505.54	-2.87
avg			*-0.42*			*-0.72*			*-1.11*			*-0.75*

bst: our best solution score, BKS: the best known solution score, **g%**: the percent deviation between our best and the BKS scores (= 100(bst-BKS)/BKS).
Sources for the BKS scores: Dominguez et al. (2014) and Fuellerer et al. (2010).
*Lower scores for these instances were mistakenly reported in Dominguez et al. (2014). This was clarified by contacting the authors of Dominguez et al. (2014).

Table A.7. Loading characteristics of the solutions obtained for the 2|O|SPD version of 2L-SPD

Inst	Class 2			Class 3			Class 4			Class 5		
	avg2D	max2D	T_RHL	avg2D	max2	T_RHL	avg2D	max2	T_RHL	avg2D	max2	T_RHL
1	58.1	85.6	80.5	55.7	87.4	78.0	54.0	90.6	56.4	55.5	82.6	87.6
2	39.8	70.1	81.2	46.1	77.5	80.3	65.5	82.9	85.0	56.0	86.6	46.7
3	48.0	76.4	48.4	59.6	91.4	49.4	67.7	91.6	71.6	69.1	94.3	83.0
4	42.9	77.4	82.0	53.4	83.9	85.1	51.6	82.0	84.3	63.3	94.3	84.0
5	47.0	89.3	43.5	64.0	94.3	52.0	47.8	94.5	56.9	61.9	95.0	83.0
6	51.3	81.1	60.1	64.5	91.4	53.9	53.0	81.3	72.5	70.1	96.8	43.2
7	59.5	90.5	22.3	74.7	91.1	38.5	54.1	93.4	24.6	51.9	97.0	24.2
8	48.0	85.3	15.4	57.7	90.4	21.0	53.9	86.1	24.2	63.7	85.3	63.2
9	45.3	85.4	72.5	70.8	94.0	26.9	72.3	94.1	21.7	72.4	95.0	56.1
10	67.2	89.5	23.7	56.3	89.4	29.5	67.3	96.5	37.7	72.3	93.5	41.0
11	54.4	90.6	12.2	61.5	92.3	12.7	70.8	96.0	13.7	70.6	93.8	30.3
12	45.6	75.5	81.0	65.4	90.8	53.2	51.9	84.1	86.1	80.3	97.4	46.0
13	54.6	84.0	11.0	62.2	96.0	35.1	74.6	95.6	13.7	67.5	99.1	55.4
14	64.3	90.0	44.8	69.2	94.1	38.6	66.8	94.1	64.4	62.9	94.3	62.3
15	66.1	96.1	28.6	73.3	94.1	19.9	77.1	96.9	11.9	68.8	98.1	12.5
16	38.5	75.8	83.5	72.3	92.5	44.0	69.5	92.1	37.8	64.2	96.6	27.0
17	60.3	87.0	26.0	66.2	92.9	49.5	62.7	94.1	18.2	83.3	98.4	39.6
18	66.8	93.8	13.8	66.8	92.9	8.7	66.1	92.6	11.3	62.1	98.6	14.3
19	67.1	94.6	17.8	68.3	91.4	31.5	72.5	94.3	7.6	84.5	96.6	24.8
20	59.6	93.9	5.0	71.3	92.9	5.1	67.9	95.4	6.2	73.3	97.8	5.9
21	63.8	95.3	16.1	65.4	93.6	14.0	63.3	95.8	19.9	73.8	98.0	17.2
22	65.4	93.1	8.1	71.5	94.6	10.6	69.6	97.1	30.3	75.6	97.9	18.9
23	64.4	94.3	25.6	71.2	94.1	21.9	64.9	93.5	17.0	74.5	96.9	34.0
24	60.4	91.0	24.8	69.1	94.5	40.9	67.2	95.4	33.5	63.4	91.4	82.6
25	62.7	92.5	4.3	71.3	97.1	9.8	73.5	96.3	17.4	64.2	98.1	72.6
26	65.1	90.8	7.6	71.1	95.5	7.4	69.4	93.6	5.0	73.1	97.9	12.5
27	62.6	93.3	20.9	71.7	93.9	15.3	72.7	97.4	18.1	76.9	98.1	32.3
28	63.2	96.4	12.1	69.6	97.4	8.1	72.6	97.9	7.1	79.2	97.8	18.0
29	67.4	94.1	6.5	70.1	96.8	5.7	72.0	97.1	5.5	77.7	98.0	5.7
30	67.5	93.3	3.6	68.8	97.3	10.7	76.7	97.3	3.0	76.3	98.9	7.6
31	66.5	96.5	3.4	71.9	97.0	2.4	73.8	98.3	13.0	78.2	99.3	6.7
32	67.6	98.5	3.7	70.6	97.0	3.2	72.9	96.6	11.0	80.4	98.9	8.4
33	64.4	98.0	13.2	72.9	97.4	7.1	75.8	98.0	14.6	76.3	98.0	7.0
34	68.5	97.9	5.7	67.3	98.4	5.6	73.2	97.9	4.1	77.3	99.0	8.6
35	66.9	97.5	8.8	73.3	98.1	7.2	74.7	97.1	7.1	78.1	98.8	17.5
36	66.8	97.4	8.2	73.3	98.3	4.7	75.3	99.0	4.1	77.8	98.6	4.2
avg	*59.1*	*89.8*	*28.5*	*66.9*	*93.4*	*27.4*	*67.1*	*93.8*	*28.2*	*71.0*	*96.0*	*35.7*

avg2D: the average loading area utilization over the solution arcs, **max2D**: the highest loading area utilization among the solution arcs, **T_RHL**: the number of routes found loading feasible over the total number of routes examined regarding their loading feasibility

Table A.8. Loading characteristics of the solutions obtained for the 2|R|SPD version of 2L-SPD

Inst	Class 2			Class 3			Class 4			Class 5		
	avg2D	max2D	T_RHL	avg2D	max2	T_RHL	avg2D	max2	T_RHL	avg2D	max2	T_RHL
1	58.1	85.6	86.4	55.7	87.4	87.2	72.6	98.0	67.8	64.6	88.1	88.9
2	39.8	70.1	98.5	51.1	93.3	92.2	66.7	87.0	93.5	40.8	86.6	47.1
3	63.6	89.3	75.7	59.9	98.6	57.8	67.7	91.6	78.4	69.1	94.3	90.0
4	42.9	77.4	94.6	58.0	95.1	92.9	48.9	98.4	92.5	63.3	94.3	90.4
5	58.9	95.1	45.9	64.0	94.3	57.5	43.0	94.5	63.1	60.9	98.5	88.9
6	61.3	95.1	73.9	64.1	97.6	58.7	53.0	81.3	83.0	70.1	96.8	45.7
7	59.7	93.5	27.0	67.6	98.1	55.0	79.0	98.4	30.5	51.9	97.0	24.2
8	48.6	91.3	19.6	59.9	94.5	24.9	55.9	86.1	28.5	66.9	99.6	63.9
9	46.9	88.4	87.5	73.4	94.9	35.6	71.2	95.4	26.1	72.4	95.0	54.2
10	67.4	89.3	27.1	61.5	96.8	31.7	69.9	98.3	37.1	70.1	93.5	42.3
11	57.1	92.1	15.9	56.6	99.3	14.8	70.3	99.0	17.5	70.6	93.8	32.0
12	46.3	81.6	97.8	63.3	94.0	62.4	51.9	84.1	91.5	76.3	99.4	55.1
13	68.2	95.1	13.7	65.6	97.1	43.6	80.4	98.0	16.3	67.5	99.1	58.3
14	68.0	93.3	56.4	70.5	97.5	46.5	66.8	94.1	68.1	77.8	99.8	64.9
15	67.9	98.6	35.8	70.6	94.6	24.8	70.7	98.6	15.2	70.7	98.8	12.4
16	38.5	75.8	94.4	76.0	95.0	56.3	69.5	92.1	35.3	61.9	96.6	29.1
17	65.6	93.4	29.7	67.3	96.4	56.8	75.9	98.3	18.3	80.2	98.4	36.6
18	65.6	93.9	18.6	72.3	97.9	9.0	66.6	97.6	12.1	64.2	99.9	15.1
19	68.3	95.1	31.7	70.5	97.3	39.3	73.8	98.1	8.1	84.3	99.8	28.6
20	66.8	94.3	7.3	74.0	96.8	4.6	68.3	99.0	6.4	78.4	99.8	5.8
21	68.6	99.0	18.7	73.0	96.8	18.2	69.3	98.5	21.7	73.6	99.6	17.8
22	71.2	96.0	9.0	73.3	96.5	10.7	71.5	98.6	33.8	77.3	99.1	17.8
23	68.6	96.6	31.9	71.8	97.8	28.0	70.3	98.0	21.9	75.4	99.0	38.2
24	68.6	97.1	32.6	69.3	97.0	53.7	66.9	99.0	41.3	59.9	97.6	84.5
25	71.2	97.1	3.5	75.0	98.9	10.0	76.9	99.4	18.2	64.1	98.1	74.5
26	69.9	98.3	6.9	71.2	97.9	7.2	73.6	99.0	5.5	77.2	98.8	12.6
27	65.8	96.0	28.1	75.6	98.6	17.9	70.5	98.8	23.0	77.4	98.1	37.7
28	62.4	98.0	11.9	71.8	99.0	11.5	76.5	98.9	6.0	78.5	99.8	18.4
29	69.7	98.4	6.0	72.7	99.4	4.5	74.6	98.6	4.5	79.6	99.8	5.1
30	71.5	99.4	3.5	74.1	98.8	10.8	77.7	99.0	3.5	83.2	99.6	6.9
31	69.4	98.3	3.5	74.7	98.5	2.7	74.6	98.8	13.3	74.0	100.0	6.5
32	71.2	98.5	3.9	74.5	99.1	3.9	74.3	99.9	11.2	79.0	99.8	7.6
33	70.4	98.9	17.3	74.2	98.8	9.6	76.9	100.0	17.0	78.2	99.6	6.6
34	73.6	98.4	6.1	72.0	98.8	5.0	75.7	99.3	4.2	79.6	99.1	7.1
35	71.3	97.5	11.6	75.8	98.4	8.8	78.7	99.4	7.1	79.8	99.8	20.3
36	70.2	98.9	8.7	76.5	98.9	4.8	76.1	99.8	4.2	79.4	100.0	4.7
	63.1	*93.2*	*34.5*	*68.8*	*96.9*	*32.2*	*69.6*	*96.5*	*31.3*	*71.6*	*97.7*	*37.2*

The notation of Table A.7 is used

Table A.9. Summary of the 2L-SPD LIFO runs

	Fixed Item Orientation				Item Rotations			
	bst	**avg**	**t**	**%g**	**bst**	**avg**	**t**	**%g**
L01-4	284.42	284.42	10.7	0.00	281.66	281.66	46.0	0.00
L02-4	285.57	285.57	40.7	0.00	285.57	285.57	27.5	0.00
L03-4	350.98	350.98	42.9	0.00	350.98	350.98	85.6	0.00
L04-4	360.13	360.13	33.4	0.00	358.20	358.20	148.0	0.00
L05-4	383.86	383.86	278.1	0.00	378.23	378.23	58.8	0.00
L06-4	391.83	391.83	621.8	0.00	380.95	380.95	107.8	0.00
L07-4	601.83	601.83	32.8	0.00	601.83	601.83	23.1	0.00
L08-4	618.40	618.40	104.1	0.00	616.89	616.89	74.5	0.00
L09-4	437.24	437.24	50.2	0.00	427.86	427.86	111.3	0.00
L10-4	599.61	599.61	534.3	0.00	597.14	597.14	640.3	0.00
L11-4	677.46	677.46	432.0	0.00	675.16	675.16	518.7	0.00
L12-4	394.15	394.15	247.0	0.00	392.06	392.06	138.7	0.00
L13-4	2311.91	2311.91	1261.4	0.00	2246.00	2257.74	7468.9	0.52
L14-4	699.56	699.77	3746.3	0.03	693.34	693.34	378.7	0.00
L15-4	917.42	922.22	6450.7	0.52	889.04	890.07	5557.2	0.12
L16-4	440.79	440.79	1061.0	0.00	433.45	433.45	1710.3	0.00
L17-4	473.57	478.34	6826.8	1.01	458.04	461.26	6264.0	0.70
L18-4	981.21	992.53	11761.5	1.15	942.45	957.08	6230.6	1.55
L19-4	624.63	628.79	8246.8	0.67	619.71	622.23	6100.3	0.41
L20-4	462.16	467.11	6776.5	1.07	438.09	443.06	5861.8	1.13
L21-4	824.20	829.95	8969.7	0.70	804.06	809.72	5690.2	0.70
L22-4	860.65	868.22	9956.2	0.88	853.20	859.67	6422.8	0.76
L23-4	858.69	868.30	8455.3	1.12	820.55	835.99	5404.0	1.88
L24-4	821.03	829.93	6776.6	1.08	799.38	805.04	5064.5	0.71
L25-4	1073.55	1083.97	9098.0	0.97	1043.15	1057.77	8867.4	1.40
L26-4	1077.06	1091.49	6964.1	1.34	1028.96	1039.95	8909.8	1.07
L01-5	273.61	273.61	70.3	0.00	272.03	272.03	35.2	0.00
L02-5	269.77	269.77	85.3	0.00	264.96	264.96	129.3	0.00
L03-5	356.54	356.54	230.2	0.00	335.19	335.19	882.3	0.00
L04-5	365.46	365.46	139.3	0.00	364.29	364.29	121.4	0.00
L05-5	365.19	365.19	353.3	0.00	362.12	362.12	954.8	0.00
L06-5	397.74	397.74	75.5	0.00	388.31	388.31	191.1	0.00
L07-5	613.67	613.67	115.7	0.00	607.70	607.70	2.8	0.00
L08-5	634.86	634.86	130.6	0.00	617.22	617.22	121.5	0.00
L09-5	462.53	464.99	1145.4	0.53	436.32	436.32	410.7	0.00
L10-5	561.70	561.70	1641.1	0.00	558.44	558.44	466.3	0.00
L11-5	546.36	546.36	136.5	0.00	519.06	519.06	2414.9	0.00
L12-5	399.25	399.25	1596.1	0.00	396.22	396.22	577.8	0.00
L13-5	2301.19	2301.19	206.3	0.00	2285.59	2286.01	1633.1	0.02
L14-5	775.26	778.16	8127.9	0.37	750.11	750.11	523.0	0.00
L15-5	904.09	931.92	5912.9	3.08	894.47	912.85	4657.5	2.06
L16-5	455.51	455.51	734.5	0.00	455.51	455.51	461.7	0.00
L17-5	520.39	523.46	10402.6	0.59	505.94	510.71	10083.2	0.94
L18-5	861.81	866.12	7613.6	0.50	859.93	860.58	3574.2	0.08
L19-5	584.11	584.47	13102.6	0.06	578.20	579.52	7398.0	0.23
L20-5	389.85	393.46	5982.8	0.92	386.18	388.19	6961.1	0.52
L21-5	752.77	759.94	16779.6	0.95	748.35	750.92	8356.6	0.34
L22-5	810.75	812.56	11809.8	0.22	799.98	805.80	9466.3	0.73
L23-5	781.51	790.24	8775.1	1.12	768.16	774.72	8858.7	0.85
L24-5	780.63	786.73	11467.2	0.78	777.19	780.54	7679.6	0.43
L25-5	979.99	991.87	12858.6	1.21	971.98	977.05	9827.1	0.52
L26-5	952.87	965.08	13937.3	1.28	928.57	938.48	12914.9	1.07
avg			*4465.6*	*0.43*			*3473.3*	*0.36*

bst: the best solution score obtained over the ten runs, **avg**: the average solution score over the ten runs, **t**: the average computational time elapsed for obtaining the final solution over the ten runs (in CPU sec), **%g:** the percent deviation between our best and average scores (= 100(avg-bst)/bst)

Table A.10. Summary of the 2L-SPD runs for the LIFO instances

	Fixed Item Orientation				Item Rotations			
	bst	**avg**	**t**	**%g**	**bst**	**avg**	**t**	**%g**
L01-4	258.11	258.11	0.5	0.00	258.11	258.11	0.4	0.00
L02-4	263.91	263.91	3.7	0.00	260.22	260.22	2.1	0.00
L03-4	332.28	332.28	10.4	0.00	328.30	328.30	7.2	0.00
L04-4	319.56	319.56	7.9	0.00	311.90	311.90	1.7	0.00
L05-4	339.44	339.44	1.7	0.00	339.44	339.44	1.0	0.00
L06-4	354.43	354.43	1.4	0.00	354.43	354.43	0.8	0.00
L07-4	527.64	527.64	10.1	0.00	527.64	527.64	3.3	0.00
L08-4	583.94	583.94	3.6	0.00	583.94	583.94	2.7	0.00
L09-4	400.62	400.62	3.6	0.00	400.62	400.62	2.5	0.00
L10-4	524.49	524.49	3.7	0.00	524.49	524.49	2.6	0.00
L11-4	629.69	629.69	8.7	0.00	629.69	629.69	9.5	0.00
L12-4	356.78	356.78	12.3	0.00	356.78	356.78	16.7	0.00
L13-4	2005.51	2005.51	14.5	0.00	2005.51	2005.51	11.3	0.00
L14-4	678.47	678.47	4.7	0.00	678.47	678.47	2.4	0.00
L15-4	838.51	838.66	273.0	0.02	838.51	838.81	229.2	0.04
L16-4	403.90	403.98	122.1	0.02	397.51	400.65	34.8	0.79
L17-4	410.27	410.27	15.2	0.00	410.27	410.27	10.8	0.00
L18-4	860.64	860.64	60.7	0.00	857.94	857.94	60.1	0.00
L19-4	551.93	553.21	125.7	0.23	549.83	549.83	80.7	0.00
L20-4	382.72	384.25	910.7	0.40	378.34	380.63	1072.7	0.61
L21-4	716.40	716.40	162.9	0.00	714.54	714.54	128.8	0.00
L22-4	754.27	760.21	243.2	0.79	750.14	757.77	265.3	1.02
L23-4	705.20	705.82	535.3	0.09	698.93	699.39	377.8	0.07
L24-4	710.84	711.22	267.4	0.05	709.66	711.34	329.1	0.24
L25-4	900.47	911.79	626.2	1.26	893.81	902.35	482.6	0.96
L26-4	824.22	825.47	314.0	0.15	818.63	822.03	625.9	0.42
L01-5	245.39	245.39	1.0	0.00	245.39	245.39	1.5	0.00
L02-5	257.86	257.86	0.5	0.00	257.86	257.86	0.2	0.00
L03-5	305.44	305.44	3.9	0.00	305.44	305.44	2.0	0.00
L04-5	336.28	336.28	1.2	0.00	321.16	321.16	0.6	0.00
L05-5	319.13	319.13	1.4	0.00	319.13	319.13	0.6	0.00
L06-5	351.30	351.30	32.6	0.00	351.30	351.30	18.7	0.00
L07-5	563.25	563.25	0.2	0.00	563.25	563.25	1.3	0.00
L08-5	530.96	530.96	5.4	0.00	530.96	530.96	3.3	0.00
L09-5	413.03	413.03	3.9	0.00	413.03	413.03	4.5	0.00
L10-5	504.79	505.40	11.1	0.12	503.17	503.17	12.9	0.00
L11-5	497.53	497.53	5.0	0.00	497.53	497.53	4.4	0.00
L12-5	360.92	360.92	13.8	0.00	360.92	360.92	16.5	0.00
L13-5	2111.04	2111.04	21.0	0.00	2111.04	2111.04	9.4	0.00
L14-5	683.98	684.04	5.8	0.01	683.98	683.98	9.9	0.00
L15-5	840.39	840.39	13.3	0.00	840.39	840.39	7.7	0.00
L16-5	422.23	422.23	17.6	0.00	422.04	422.04	30.0	0.00
L17-5	446.51	446.51	203.1	0.00	443.84	444.05	176.2	0.05
L18-5	792.60	792.60	236.2	0.00	786.42	787.66	50.5	0.16
L19-5	524.27	524.27	48.2	0.00	524.27	524.27	43.0	0.00
L20-5	360.59	362.36	599.3	0.49	359.14	359.37	322.1	0.07
L21-5	655.97	656.29	1361.8	0.05	655.97	656.71	875.1	0.11
L22-5	685.02	691.43	1312.5	0.94	680.04	682.57	802.7	0.37
L23-5	686.04	690.96	308.4	0.72	685.58	690.29	310.3	0.69
L24-5	692.38	699.91	361.0	1.09	691.99	700.31	344.0	1.20
L25-5	844.53	846.97	732.7	0.29	844.53	848.03	562.2	0.41
L26-5	789.18	791.59	877.7	0.31	782.35	786.83	547.1	0.57
avg			*190.8*	*0.13*			*152.3*	*0.15*

bst: the best solution score obtained over the ten runs, **avg**: the average solution score over the ten runs, **t**: the average computational time elapsed for obtaining the final solution over the ten runs (in CPU sec), **%g:** the percent deviation between our best and average scores (= 100(avg-bst)/bst)

Table A.11. Loading feasibility examinations for the LIFO and basic 2L-SPD model (Fixed Item Orientation)

	2|O|SPD-L							2|O|SPD							
Instance	F_m	F_{smd}	F_r	F_{rh}	F_a	F_p	F_{ps}	F_m	F_{smd}	F_r	F_{rh}	F_a	F_{ah}	F_p	F_{ps}
L01-4	1152.9	301.3	985.9	823.5	236.1	378.5	206.2	182.2	27.1	212.5	157.1	346.0	325.0	17.5	16.2
L02-4	2853.4	836.4	2340.3	1916.3	510.9	526.8	287.9	483.6	84.5	515.2	367.8	404.4	348.4	33.1	27.6
L03-4	1454.4	526.4	1102.5	922.7	256.3	322.1	193.6	522.6	121.9	502.8	355.8	526.0	452.2	43.7	37.7
L04-4	1240.3	452.4	945.0	795.7	219.1	327.2	192.5	266.6	51.6	279.3	192.3	467.3	424.2	29.8	26.0
L05-4	1891.1	689.0	1423.3	1101.4	384.7	412.4	243.4	581.2	186.3	483.9	328.5	474.9	448.8	19.2	17.8
L06-4	2369.5	1018.8	1521.7	1208.1	453.2	667.1	373.4	207.3	43.3	220.2	163.6	328.0	292.3	28.8	26.5
L07-4	671.9	187.4	583.7	463.8	166.8	223.8	127.1	288.3	29.1	323.3	162.7	1159.7	953.5	157.7	150.1
L08-4	1641.7	529.8	1248.2	853.0	504.8	655.2	345.8	346.2	52.1	377.0	205.5	779.2	701.4	57.0	52.9
L09-4	1376.5	586.6	940.5	789.8	205.0	301.8	154.9	180.8	38.6	199.2	140.9	391.4	318.1	60.9	58.0
L10-4	3226.6	1706.1	1652.8	1099.8	780.1	1001.8	601.3	303.3	110.9	232.5	151.8	596.7	537.3	46.6	43.3
L11-4	2204.6	1363.4	974.6	773.3	274.7	331.0	184.1	380.0	108.5	328.2	188.3	721.2	577.5	94.3	84.7
L12-4	2091.5	852.6	1434.0	997.8	600.1	849.6	433.0	255.5	55.7	260.4	128.3	1067.5	868.2	153.7	145.2
L13-4	4022.5	1712.6	2542.8	1388.0	1499.3	1950.5	966.3	309.4	66.4	297.7	170.6	893.9	765.2	95.8	90.1
L14-4	3697.9	1910.4	1939.4	947.9	1418.6	2090.4	1114.9	174.6	25.0	200.2	97.7	1146.4	965.0	158.6	155.2
L15-4	3218.9	1418.3	2074.1	1330.1	922.9	1049.6	569.1	304.3	75.2	306.9	162.9	954.2	769.4	131.3	122.3
L16-4	1881.2	870.1	1158.7	775.3	505.8	664.7	338.5	218.8	58.2	218.3	115.6	796.7	656.2	106.5	99.5
L17-4	5306.1	2481.0	3185.8	1623.7	2043.3	2656.8	1383.3	317.5	96.9	289.4	150.1	1042.9	844.4	138.1	127.9
L18-4	6421.3	3831.2	2903.0	1733.5	1514.1	1817.8	970.2	1066.3	464.3	733.7	457.3	907.6	744.7	107.1	98.4
L19-4	3808.3	2271.6	1751.8	1120.5	795.1	1043.3	487.3	238.9	86.8	218.2	137.4	632.4	459.8	134.5	128.0
L20-4	8954.1	6570.0	2833.6	1979.7	1145.6	1369.3	756.6	1399.1	842.1	711.9	402.4	1108.3	757.0	214.9	194.0
L21-4	6665.7	4577.1	2403.2	1457.6	1279.0	1831.8	986.0	346.7	166.2	268.0	149.6	1024.2	831.6	153.1	146.0
L22-4	5516.6	3834.3	1931.1	1239.7	881.4	1117.1	582.5	266.7	123.2	224.3	135.9	721.1	472.9	203.6	196.9
L23-4	10148.1	6904.1	3755.6	2497.3	1501.3	1507.9	779.8	953.6	510.3	589.0	355.3	1200.5	940.6	157.6	140.4
L24-4	5022.0	3384.8	1916.1	1297.9	793.9	1005.4	512.6	350.4	165.4	270.7	160.3	845.3	581.1	202.1	192.7
L25-4	8568.4	6487.9	2487.8	1621.3	1124.5	1393.4	725.8	439.0	251.7	313.8	185.9	1006.5	691.0	233.1	220.6
L26-4	12337.6	9721.5	3122.9	2120.5	1196.7	1051.4	556.3	849.6	579.7	396.3	241.6	877.6	751.2	90.8	84.2
L01-5	1258.3	339.4	1089.0	948.2	217.3	323.6	193.9	246.9	25.6	284.9	186.1	496.4	459.4	30.6	27.9
L02-5	936.0	220.8	870.4	747.9	170.2	233.8	151.0	255.4	39.1	292.0	208.5	306.4	297.4	7.7	6.6
L03-5	3684.0	1186.1	2878.8	1875.1	1277.9	1475.1	877.1	385.8	74.4	392.9	224.5	762.0	624.5	87.5	75.3
L04-5	1951.1	710.1	1411.5	1139.3	343.7	361.1	200.8	418.6	110.4	392.8	289.3	362.5	278.5	58.5	52.0
L05-5	2739.1	875.7	2129.2	1354.4	1045.0	1392.9	708.0	246.2	42.2	262.0	166.5	524.2	421.2	89.3	87.1
L06-5	1790.4	755.9	1176.8	933.0	345.0	444.8	261.6	268.2	65.9	259.7	170.8	463.4	387.0	62.8	58.9
L07-5	760.6	350.1	482.7	421.4	88.0	132.7	74.6	195.5	31.9	215.7	137.0	614.9	495.7	107.9	106.0
L08-5	1514.0	461.8	1208.2	863.2	468.3	580.0	343.6	391.9	53.5	414.1	213.0	1155.9	984.7	133.3	126.5
L09-5	4161.9	2116.1	2277.6	1477.7	1033.4	1164.2	686.4	389.8	116.7	340.6	229.6	473.5	333.9	112.9	107.5
L10-5	1677.0	679.6	1166.8	835.2	429.4	470.7	252.8	482.2	179.1	350.3	234.5	733.0	636.1	82.5	80.3
L11-5	1230.0	533.8	808.5	572.5	340.2	450.1	271.4	270.5	60.4	250.1	163.0	670.2	601.0	60.8	59.1
L12-5	2399.7	1030.5	1576.5	987.8	770.3	1013.3	526.7	183.2	39.3	192.8	103.3	779.0	602.8	157.1	154.0
L13-5	1549.8	795.7	895.4	711.6	232.3	243.8	132.4	250.8	62.0	253.6	140.9	581.7	426.7	140.5	138.3
L14-5	1223.5	455.2	953.6	654.5	434.8	574.4	327.7	196.3	31.2	201.9	92.5	1226.9	967.9	229.7	226.3
L15-5	1861.5	897.6	1126.6	805.8	434.7	580.4	353.2	211.3	47.1	232.5	148.9	713.7	596.5	106.5	104.8
L16-5	3358.9	1582.5	1997.4	1230.0	1015.9	1239.8	671.9	292.1	89.0	262.9	143.3	783.5	580.9	167.7	161.7
L17-5	9357.2	5453.7	4333.0	2018.4	2781.7	2974.3	1607.6	308.0	100.0	273.2	146.0	883.4	629.4	211.7	204.6
L18-5	3065.0	1703.5	1616.6	1159.6	585.8	620.3	335.2	628.2	227.8	514.5	306.5	822.4	652.8	133.9	130.0
L19-5	4229.3	2439.5	1991.5	1060.3	1234.3	1671.2	793.1	185.8	54.4	187.4	102.6	831.4	552.8	248.4	244.2
L20-5	4212.1	3027.5	1391.8	874.8	670.1	815.9	448.9	411.8	210.1	270.7	159.8	799.7	605.2	158.2	153.6
L21-5	7524.2	4597.3	3337.7	1649.9	2155.5	2608.8	1289.5	606.4	287.5	434.7	228.4	1529.7	941.7	437.5	421.8
L22-5	6889.5	4484.4	2755.0	2000.9	939.8	1036.3	491.6	569.4	294.6	390.1	233.3	1065.0	680.3	294.8	284.0
L23-5	5273.5	3535.7	2044.7	1346.5	918.1	1189.5	575.9	222.0	90.7	214.5	122.6	893.0	604.6	251.2	246.5
L24-5	4802.7	3130.0	1943.1	1173.5	1030.4	1338.4	699.6	244.1	103.2	219.1	121.3	914.1	598.9	275.0	270.3
L25-5	6839.8	4890.4	2355.1	1412.5	1208.8	1467.1	739.7	276.1	131.4	259.1	148.1	1108.2	736.9	317.3	311.2
L26-5	17776.9	12981.3	5406.7	2704.5	3224.7	3463.5	1479.5	390.7	228.1	270.6	158.2	944.0	751.0	169.9	166.8

F_m: number of loading feasibility for local-search moves, F_{smd}: move feasibility retrievals from the SMD instances, F_r: number of necessary route loading feasibility examinations, F_{rh}: number of route feasibility retrievals from the route hashtables, F_a: number of necessary arc loading feasibility examinations, F_{ah}: number of arc feasibility retrievals from the arc hashtables, F_p: calls to the packing heuristic procedure, F_{ps}: number of successful packing heuristic calls.
Note: Reported values are divided by 10^3.

Table A.12. Loading feasibility examinations for the LIFO and basic 2L-SPD model (Item Rotations)

	2|R|SPD-L							2|R|SPD							
Instance	F_m	F_{smd}	F_r	F_{rh}	F_a	F_p	F_{ps}	F_m	F_{smd}	F_r	F_{rh}	F_a	F_{ah}	F_p	F_{ps}
L01-4	969.4	269.2	824.2	692.1	190.9	305.4	165.3	175.9	25.6	206.5	153.3	336.1	315.7	18.1	17.1
L02-4	2625.9	743.6	2197.7	1792.0	493.3	499.5	284.3	377.9	65.6	411.1	288.5	363.4	316.9	32.4	28.0
L03-4	1488.7	534.0	1129.3	933.5	280.8	346.0	217.9	470.9	107.7	455.1	310.8	544.0	474.8	49.1	44.1
L04-4	1074.7	381.7	837.7	705.5	195.7	287.5	174.2	217.2	38.8	234.8	159.8	442.1	405.5	30.0	27.8
L05-4	1496.0	567.6	1085.8	876.0	251.2	264.1	162.6	486.7	140.4	426.2	292.4	421.9	399.7	18.1	17.1
L06-4	1514.0	569.5	1082.6	893.0	268.6	381.9	220.6	192.1	37.9	208.6	154.5	323.2	288.6	29.9	28.2
L07-4	577.0	147.1	514.5	414.0	139.8	173.0	103.8	272.0	24.9	309.4	156.1	1112.3	935.9	150.2	145.9
L08-4	1269.9	428.3	949.9	662.3	370.2	465.6	259.1	336.3	50.3	366.6	203.3	758.8	683.5	60.8	57.7
L09-4	1076.3	465.9	725.9	609.8	167.0	250.3	142.7	174.6	36.3	194.5	135.8	403.9	329.4	64.4	62.1
L10-4	2955.9	1636.0	1438.4	997.3	613.2	753.3	467.0	269.7	94.9	213.0	141.2	568.6	515.6	45.1	42.9
L11-4	1949.1	1225.6	847.3	708.2	191.8	228.8	132.5	355.8	95.7	315.3	183.4	706.7	566.9	103.9	96.8
L12-4	1688.1	641.9	1223.4	826.2	543.1	742.5	394.0	231.8	50.9	238.5	119.0	1006.6	826.4	152.2	146.9
L13-4	3431.1	1505.9	2128.5	1141.9	1308.8	1659.2	886.1	280.1	58.6	273.0	156.2	887.5	768.1	97.4	93.6
L14-4	2372.3	1078.7	1416.2	691.9	1002.2	1388.7	768.0	165.3	23.0	192.1	96.2	1103.8	936.3	154.5	152.6
L15-4	3500.8	1398.4	2370.9	1235.7	1405.4	1519.4	873.9	277.0	65.6	285.2	151.1	925.0	748.4	142.7	136.7
L16-4	1995.9	916.0	1238.4	787.2	591.4	737.5	397.7	209.3	53.4	212.7	111.5	806.5	669.9	112.1	107.1
L17-4	4125.6	1927.6	2513.1	1303.3	1605.6	2024.3	1122.7	263.7	76.8	250.7	127.1	985.1	807.6	140.5	133.9
L18-4	5500.5	3416.0	2366.4	1495.1	1111.6	1235.6	674.8	922.8	363.0	693.4	412.6	967.1	806.5	117.0	110.2
L19-4	3158.9	1888.3	1452.4	964.8	618.6	775.9	390.4	215.6	73.8	205.2	132.7	586.5	432.5	128.4	124.1
L20-4	7798.7	5754.7	2413.2	1670.4	992.3	1137.2	662.5	1380.7	829.9	709.6	410.4	1033.1	747.5	201.4	187.8
L21-4	4914.5	3358.1	1826.5	1184.4	885.8	1216.0	705.0	288.6	127.5	248.2	142.3	950.8	790.9	136.6	132.3
L22-4	5698.1	3977.5	1976.2	1175.4	1033.5	1302.4	701.6	222.4	93.8	206.6	122.0	716.4	468.7	218.3	213.9
L23-4	8369.8	5618.2	3168.4	2114.6	1287.0	1292.0	733.1	748.0	394.0	486.8	301.8	1035.5	838.0	142.3	132.5
L24-4	3882.8	2578.5	1542.0	1097.8	570.8	680.4	365.0	294.3	130.2	249.6	143.4	840.3	579.7	215.2	208.5
L25-4	8109.3	6137.9	2353.9	1590.4	997.3	1174.9	655.5	415.8	240.0	298.8	176.9	970.0	678.2	233.4	224.5
L26-4	13040.7	10281.0	3268.7	2006.3	1520.3	1427.7	761.2	942.9	634.1	443.6	260.9	1003.8	815.4	134.0	125.4
L01-5	1057.8	274.6	943.9	823.2	187.5	255.5	166.7	232.1	23.2	270.1	177.9	488.5	453.6	31.2	29.4
L02-5	844.9	195.6	790.2	680.2	152.3	197.4	134.1	246.8	37.5	283.9	204.0	302.3	293.5	8.0	7.3
L03-5	3099.1	1075.0	2327.3	1490.8	1089.1	1219.0	769.9	310.9	57.6	323.9	193.2	639.1	530.5	80.7	73.2
L04-5	1602.0	591.3	1161.7	935.2	287.4	292.3	173.7	367.7	97.2	345.3	250.7	338.1	266.6	57.9	54.0
L05-5	1897.0	587.7	1516.1	1032.9	665.0	874.8	472.0	252.4	41.2	269.2	166.3	560.5	456.9	95.3	93.9
L06-5	1459.0	576.1	1023.5	800.2	311.0	393.4	229.5	261.1	65.3	248.6	169.2	433.7	365.1	59.7	57.0
L07-5	355.3	88.9	323.9	274.4	71.9	107.9	60.2	192.6	31.4	212.4	135.5	611.0	496.9	107.7	106.7
L08-5	1282.6	392.6	1038.4	713.2	441.9	555.2	322.5	357.7	44.8	383.7	197.4	1114.1	950.6	142.6	138.6
L09-5	3149.7	1487.1	1866.6	1174.5	888.5	948.2	597.4	338.1	97.1	306.0	211.5	424.0	307.2	101.8	98.5
L10-5	1528.4	647.5	1023.9	731.7	376.4	393.9	218.2	445.9	158.7	329.9	229.4	695.2	621.4	69.6	68.6
L11-5	1169.9	478.0	802.9	579.0	322.1	404.5	254.7	260.8	56.8	242.7	158.6	666.3	602.5	60.0	59.1
L12-5	2062.6	874.9	1375.1	862.5	673.8	840.0	459.9	174.3	37.2	184.4	97.1	779.2	608.0	163.2	161.8
L13-5	1133.8	549.1	699.9	590.0	142.5	140.3	85.2	251.1	61.1	254.2	139.1	586.1	431.3	147.1	146.0
L14-5	2098.9	797.0	1483.9	816.7	915.4	1154.1	653.9	182.7	26.4	191.9	88.6	1177.1	937.6	228.0	226.5
L15-5	1468.1	645.9	968.4	691.9	374.4	475.7	307.7	209.1	46.7	231.0	155.1	655.8	563.5	88.9	88.3
L16-5	2021.9	895.1	1280.0	854.4	576.1	689.6	409.6	257.6	75.0	238.3	128.8	751.2	565.4	165.7	162.1
L17-5	6768.8	3847.4	3244.0	1570.8	2068.1	2259.7	1314.0	282.4	84.7	262.1	137.8	890.2	645.0	224.6	221.2
L18-5	2492.7	1327.3	1399.8	1002.1	506.3	522.5	297.0	539.7	190.9	459.3	284.5	733.7	594.0	127.0	125.5
L19-5	3029.8	1706.7	1487.1	838.6	863.9	1076.7	571.4	174.9	49.7	181.0	99.1	811.2	534.1	262.0	260.0
L20-5	3317.0	2390.3	1113.7	738.2	520.1	652.7	404.6	340.6	165.3	241.4	141.8	761.2	576.1	168.3	166.1
L21-5	5147.0	3281.5	2174.6	1194.7	1278.3	1419.3	819.1	485.3	224.6	367.0	192.6	1372.9	872.2	434.2	427.4
L22-5	6110.4	3997.5	2499.3	1641.3	1061.7	1042.1	552.7	576.8	290.3	402.8	227.7	1147.4	727.0	362.3	355.7
L23-5	3822.3	2493.9	1596.8	1065.9	706.6	866.0	474.4	193.4	73.6	202.1	117.3	836.8	577.5	242.0	239.7
L24-5	4597.9	2990.7	1871.0	1096.9	1036.9	1311.8	708.1	219.2	87.2	209.2	111.9	926.1	602.2	299.6	296.8
L25-5	4960.4	3544.1	1765.8	1139.1	836.1	994.4	563.9	242.0	107.0	248.2	141.5	1093.2	756.6	313.8	311.1
L26-5	9414.8	7352.0	2500.5	1594.0	1118.1	1054.3	583.9	577.1	336.7	366.9	204.1	1248.1	965.1	254.5	251.3

The notation of Table A.11 is used