An exact approach for the vehicle routing problem with two-dimensional loading constraints

Manuel Iori [†], Juan-José Salazar-González [‡], Daniele Vigo [†]

[†] *DEIS, Dipartimento di Elettronica, Informatica e Sistemistica, Università degli Studi di Bologna, Viale Risorgimento 2, 40136 Bologna, Italy*

E-mails: miori@deis.unibo.it, dvigo@deis.unibo.it

[‡] *DEIOC, Departamento de Estadistíca, Investigación Operativa y Computación, Universidad de La Laguna, 38271 La Laguna, Tenerife, Spain*

E-mail: jjsalaza@ull.es

December 7, 2004; Revised July 29, 2005; 2nd revision April 19, 2006

Abstract

We consider a special case of the symmetric capacitated vehicle routing problem, in which a fleet of K identical vehicles must serve n customers, each with a given demand consisting in a set of rectangular two-dimensional weighted items. The vehicles have a two-dimensional loading surface and a maximum weight capacity. The aim is to find a partition of the customers into routes of minimum total cost and such that, for each vehicle, the weight capacity is taken into account and a feasible two-dimensional allocation of the items into the loading surface exists.

The problem has several practical applications in freight transportation and it is $\mathcal{NP}$-Hard in the strong sense. We propose an exact approach, based on a branch-and-cut algorithm, for the minimization of the routing cost, that iteratively calls a branch-and-bound algorithm for checking the feasibility of the loadings. Heuristics

are also used in order to improve the overall performance of the algorithm. The effectiveness of the approach is shown by means of computational results.

1 Introduction

Given a central depot, the Vehicle Routing Problem (VRP) calls for the determination of the optimal set of routes to be performed by a fleet of vehicles, in order to satisfy the demand of a given set of customers. Several important variants of this basic problem were extensively studied in the literature (see, e.g., Toth and Vigo (2002b) for a recent review). In particular, the symmetrical Capacitated VRP (CVRP) is the well-known variant of the VRP where all vehicles are identical and have a maximum loading capacity, and all the arcs in the graph representing the underlying road network can be travelled along both directions, producing the same cost.

In the CVRP literature, the demand of each customer is generally expressed by a positive integer that represents the total weight or volume of the demanded items. In this case, checking the feasibility of a solution simply requires one to ensure that the sum of the demands of the customers assigned to each vehicle does not exceed its total loading capacity. However, in many practical freight distribution applications, the loading of the items into the vehicles can represent a difficult problem, especially for large-size items. In this case, the loading pattern of the items on each vehicle must be found in order to achieve feasible solutions to the routing problem. These loading issues may have a great impact on the sequencing of the customers along the routes, and common examples of such situations may be found in the distribution of goods such as furniture, mechanical components and household appliances.

Many of these applications incorporate relevant additional features that influence the actual loading problem to be solved. For example, the items are often transported on top of rectangular bases, e.g., large pallets of suitable size, and due to their fragility or shape, they

may not be stacked one over the other. In this case, the general three-dimensional loading problem reduces to a suitably defined two-dimensional loading problem of the rectangular item bases on the vehicle loading surface. It is also frequent that, due to the structure of the pallet or to the non-uniform item weight distribution, the items may not be picked-up from any side by the most common loading/unloading equipment, such as forklift trucks. For these reasons, we assume in this paper that items have a fixed loading/unloading orientation. We note here that, although the solution approach we propose may be adapted to allow item rotation, this additional feature would make the problem considerably more difficult to solve (see, e.g., Dell'Amico et al. (2002) and Boschetti and Mingozzi (2003b)).

Vehicles are generally rear-loaded and load rearrangement at the customer sites can be difficult, time-consuming or even impossible, due to the weight and size of the items. In addition, forklift trucks are not normally able to perform significant lateral shifts in the loading and unloading process of an item, therefore each item to be unloaded must not be blocked by other items yet to be unloaded and laying, even partially, in the rectangular area from its loading position to the loading/unloading side of the truck. Finally, when the demand of a customer is made up by more than one item, all such items should be assigned to the same vehicle in order to avoid split deliveries.

In this paper we investigate a variant of the CVRP with the above-described two-dimensional additional loading constraints, hereafter denoted as 2L-CVRP. In the 2L-CVRP all vehicles are identical, have a known weight capacity and a single rectangular loading surface that may be accessed only from one side. The demand of each customer is defined by a set of rectangular items with given size, orientation and weight. All the items of a given customer must be assigned to a single vehicle. We also assume that the unloading of the items of a customer must not be blocked by items of customers to be visited later along the route. In the following, these two latter requirements will be referred to as the *item clustering* and the *sequential loading* constraints, respectively.

The 2L-CVRP combines the classical and extensively studied CVRP, with a loading problem that is closely related to the well-known Two-Dimensional Bin Packing Problem (2BPP). The 2BPP calls for the determination of a packing pattern of a given set of rectangular items into the minimum number of identical rectangular bins.

It is clear that the 2L-CVRP is strongly $\mathcal{NP}$-Hard since it generalizes the CVRP. Moreover, the two problems which are combined to obtain the 2L-CVRP are extremely difficult to solve. State-of-the-art exact approaches for the CVRP have solved instances with up to 135 customers, but within a reasonable computing time (i.e., some hours on a common PC) they can generally solve instances with up to 100 customers (see, e.g., Toth and Vigo (2002b), and Fukasawa et al. (2004)).

Also the 2BPP is very difficult to solve in practice. Exact algorithms and lower bounds were recently proposed by Martello and Vigo (1998), Fekete and Schepers (2001, 2004, 2006), Boschetti and Mingozzi (2003a, 2003b) and Pisinger and Sigurd (2006). Exact approaches for the 2BPP are generally able to solve instances with up to 100 items, but fail in many cases for smaller instances. In the literature, the problems derived from the 2BPP with some additional side constraints are generally included in the category of Container Loading Problems (2CLP). For the 2CLP, heuristic approaches have been proposed by Pisinger (1998, 2002) and Bortfeld and Gehring (2000), an analytical model was proposed by Chen et al. (1995) and an LP-based bound was presented by Scheithauer (1992, 1999). Finally, packing requirements similar to the sequential loading we consider are studied in Pisinger et al. (2005).

The 2L-CVRP has not been previously studied in the literature. The only closely-related reference we are aware of is on a VRP with three-dimensional loading constraint introduced by Türkay (2003), who proposed a general integer programming model derived from the Container Loading model proposed by Chen et al. (1995). The resulting approach was used to solve an instance involving 5 items and 5 customers. In addition, pickup and

delivery problems with rear-loaded vehicles were also studied in the literature, see, e.g., Xu et al. (2003).

This paper presents an exact algorithm for the solution of the 2L-CVRP based on a branch-and-cut approach, that proved to be a successful technique for the solution of related problems such as the CVRP (see, e.g., Naddef and Rinaldi (2002)). Within the algorithm we used both basic classes of valid inequalities derived for the CVRP, as well as specific valid inequalities associated with infeasible loading sequences. To this end, the feasibility of a given loading pattern is determined through lower bounds, effective heuristics and a specialized branch-and-bound algorithm. The overall algorithm was extensively tested on benchmark instances derived from the CVRP test problems and showed a satisfactory behavior, being able to optimally solve instances with up to 35 customers and more than 100 items.

The paper is organized as follows. Section 2 describes in detail the new problem and introduces the necessary notation. Section 3 presents an exact approach derived from the standard branch-and-cut algorithm for the CVRP, strengthened by new inequalities that take into account the vehicle loading component of the 2L-CVRP. The separation of these loading constraints is addressed in Section 4 through a specific branch-and-bound procedure. Section 5 examines the computational results and Section 6 draws some conclusions.

2 Problem Description

The 2L-CVRP may be defined as follows. We are given a complete undirected graph $G = (V, E)$, in which V defines the set of $n+1$ vertices corresponding to the depot (vertex 0) and to the customers (vertices $1, \ldots, n$). For each edge $e \in E$ the associated travelling cost, c_e, is defined. In the following, a given edge e can be also represented by its endpoint vertices $\{ij\}$.

A set of K identical vehicles is available at the depot. Each vehicle has a weight capacity

D and a rectangular loading surface, that is accessible from a single side for loading and unloading operations, whose width and height are equal to W and H, respectively. We also denote by $A = WH$ the total area of the loading surface.

Each customer i ($i = 1, \ldots, n$) is associated with a set of m_i rectangular items, whose total weight is equal to d_i, and each having specific width and height equal to $w_{i\ell}$ and $h_{i\ell}$, ($\ell = 1, \ldots, m_i$), respectively. Each item will be denoted by a pair of indices (i, ℓ). In addition, we denote by $a_i = \sum_{\ell=1}^{m_i} w_{i\ell} h_{i\ell}$ the total area of the items of customer i. Finally, let $M = \sum_{i=1}^{n} m_i$ denote the total number of items. We assume that all the above input data are positive integers.

The loading of the items on a vehicle is subject to the following specific constraints:

Item clustering: All the items of a given customer must be loaded on the same vehicle. Therefore, for each customer i, we assume that $d_i \leq D$ and there exists a feasible two-dimensional loading of the m_i items into a single vehicle surface ($i = 1, \ldots, n$).

Item orientation: Items have a fixed orientation (i.e., they may not be rotated) and must be loaded with their sides parallel to the sides of the loading surface (i.e., a so-called *orthogonal loading* is required).

Sequential loading: Unloading operations at the customers' site are performed through a single side of the vehicle and may not result in a load rearrangement on the vehicle or in lateral shifts of the item to be unloaded. Therefore, when unloading the items of a customer, no item belonging to customers served later along the route may either be moved or block the removal of the items of the current customer. In other words, no lateral movement of the items is allowed inside the vehicle when serving a customer.

We now give a more formal definition of a *feasible route* in our routing problem. A feasible route is associated with a subset S of customers and with a sorting denoted by a

bijection $\sigma : S \to \{1, \ldots, |S|\}$, where $\sigma(i)$ is the order in which customer $i \in S$ is visited along the route. A first condition that a route (S, σ) must satisfy is relative to the item weights, and is the classical capacity restriction of the CVRP:

Condition 1: The weight capacity of the vehicle may not be violated, i.e.,

$$\sum_{i \in S} d_i \leq D.$$

The remaining conditions refer to the loading pattern. It is convenient to map the loading surface onto the positive quadrant of a Cartesian coordinate system, whose origin $(0, 0)$ corresponds to the bottom-left corner of the loading surface in the vehicle, and where the x-axis and y-axis correspond to its bottom and left sides, respectively. The position of an item (i, ℓ) within the vehicle can be defined by two variables $x_{i\ell}$ and $y_{i\ell}$, representing the coordinates of its bottom-left corner in the loading surface. The loading/unloading side of the vehicle is placed at height H. Then, the values for these variables must satisfy the following conditions:

Condition 2: The items must be completely contained within the loading surface, i.e.,

$$0 \leq x_{i\ell} \leq W - w_{i\ell} \qquad \text{and} \qquad 0 \leq y_{i\ell} \leq H - h_{i\ell}$$

for all $i \in S$ and $\ell \in \{1, \ldots, m_i\}$.

Condition 3: No two items can overlap, i.e.,

$$x_{i\ell} + w_{i\ell} \leq x_{j\ell'} \quad \text{or} \quad x_{j\ell'} + w_{j\ell'} \leq x_{i\ell} \quad \text{or} \quad y_{i\ell} + h_{i\ell} \leq y_{j\ell'} \quad \text{or} \quad y_{j\ell'} + h_{j\ell'} \leq y_{i\ell}$$

for all $i, j \in S$, $\ell \in \{1, \ldots, m_i\}$, $\ell' \in \{1, \ldots, m_j\}$, and $(i, \ell) \neq (j, \ell')$.

Condition 4: (Sequential loading) When unloading an item, say (i, ℓ), no item of customers served later along the route may lay, even partially, in the rectangular area from item (i, ℓ) and the loading/unloading side of the truck, i.e.,

$$y_{i\ell} \geq y_{j\ell'} + h_{j\ell'} \quad \text{or} \quad x_{i\ell} + w_{i\ell} \leq x_{j\ell'} \quad \text{or} \quad x_{j\ell'} + w_{j\ell'} \leq x_{i\ell}$$

for all $i, j \in S : \sigma(i) < \sigma(j)$, $\ell \in \{1, \ldots, m_i\}$ and $\ell' \in \{1, \ldots, m_j\}$.

The *cost* of a feasible route is the sum of the travelling costs of the edges required to visit the customers in the specified sequence, starting and ending at the depot. Then, the 2L-CVRP consists of finding K feasible routes with minimum total cost, and such that each customer is exactly in one of them.

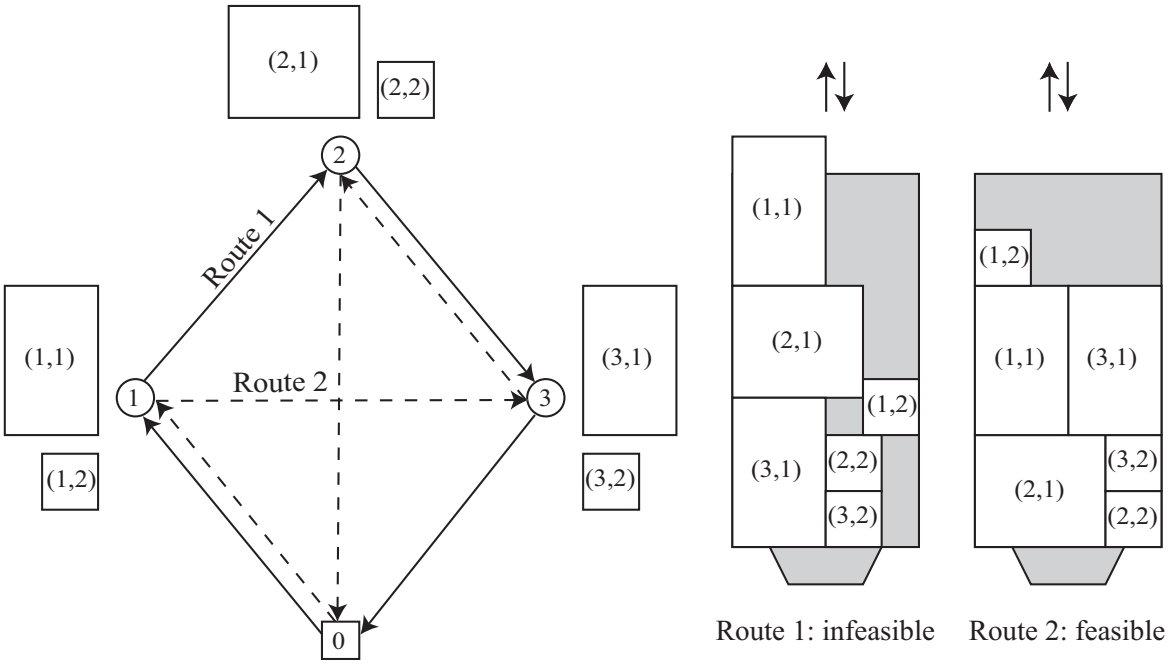

Figure 1: Feasible and infeasible routes for the 2L-CVRP.

An important observation when comparing the 2L-CVRP and the classical CVRP, is that the sequence σ is relevant to the feasibility of a route visiting customers in S. Figure 1 illustrates two different sequences σ^1 and σ^2 associated with the same customer subset $S = \{1, 2, 3\}$. Each customer has just one item. The sizes of the items and of the loading surface are proportional to those depicted in the figure. Route (S, σ^1) is defined by

$\sigma^1(1) = 1$, $\sigma^1(2) = 2$, $\sigma^1(3) = 3$, and is infeasible, since item $(2, 1)$ cannot be placed side by side with items $(1, 1)$ or $(3, 1)$. Instead, route (S, σ^2), defined by $\sigma^2(1) = 1$, $\sigma^2(2) = 3$, $\sigma^2(3) = 2$, leads to a feasible loading.

Figure 1 also shows that route (S, σ^{2R}) in the opposite direction of σ^2, defined by $\sigma^{2R}(1) = 2$, $\sigma^{2R}(2) = 3$, $\sigma^{2R}(3) = 1$, is also a feasible route for the 2L-CVRP. This is a general result, because whenever (S, σ) is a feasible route, then also (S, σ^R), where $\sigma^R(i) = |S| - \sigma(i) + 1$, is a feasible route. The loading pattern can be obtained by applying a symmetry operation with respect to line $y = H/2$, and consequently changing the coordinates (x, y) to (x^R, y^R), such that $x^R_{i\ell} = x_{i\ell}$ and $y^R_{i\ell} = H - y_{i\ell} - h_{i\ell}$. Note that, as a consequence, the opposite route of an infeasible route is also infeasible.

In the following sections we describe the overall approach used to solve the 2L-CVRP.

3 A Branch-and-Cut Approach

This section proposes an integer linear programming model for the 2L-CVRP using large families of linear inequalities. The model is later used to solve the problem through a branch-and-cut algorithm. This section also presents the separation procedures used to find violated inequalities of the families describing the model, and a primal heuristic to speed up the solution of the algorithm.

Given a customer subset S, we denote by $\delta(S)$ the set of edges with one endpoint in S and the other in $V \setminus S$. As usual, $\delta(i)$ is used instead of $\delta(\{i\})$. Moreover, given a feasible route (S, σ) we denote by $E(S, \sigma)$ the set of edges in such a route. Finally, given a customer subset S we denote by $\Sigma(S)$ the collection of sequences σ such that (S, σ) is a feasible route.

3.1 Problem Formulation

To model the 2L-CVRP we adapted the classical two-index vehicle flow model for the CVRP (see, e.g., Toth and Vigo (2002a)). It uses a binary variable z_e for each $e \in E$ which takes value 1 if and only if a vehicle travels along edge e in a feasible route. The resulting formulation is:

$$\min \sum_{e \in E} c_e z_e \tag{1}$$

$$\sum_{e \in \delta(i)} z_e = 2 \qquad \text{for all } i \in V \setminus \{0\} \tag{2}$$

$$\sum_{e \in \delta(0)} z_e = 2K \tag{3}$$

$$\sum_{e \in \delta(S)} z_e \geq 2r(S) \qquad \text{for all } S \subseteq V \setminus \{0\}, S \neq \emptyset \tag{4}$$

$$\sum_{e \in E(S,\sigma)} z_e \leq |S| - 1 \qquad \text{for all } (S, \sigma) \text{ such that } \sigma \notin \Sigma(S) \tag{5}$$

$$z_e \in \{0, 1\} \qquad \text{for all } e \in E. \tag{6}$$

Constraints (2) and (3) impose degree requirements at the customer vertices, and at the depot, respectively. Constraints (4), known as the *capacity-cut constraints*, impose both solution connectivity and feasibility with respect to Conditions 1–3 of Section 2. In these constraints, $r(S)$ is the minimum number of vehicles that are necessary to serve the customers in set S, and not considering the actual sequence in which they are visited (i.e., by disregarding Condition 4 introduced in Section 2). Constraints (5), known as *infeasible-path constraints*, consider the loading sequence requirements of Condition 4 by eliminating non-feasible sequences σ associated with a customer set S. The asymmetric variant of Constraints (5), where the edges are replaced by arcs, has been used by other authors in similar routing problems. Depending on the problem, the asymmetric infeasible-path constraint may be lifted leading to the so-called *tournament constraints*. See, e.g., Ascheuer et al. (2000, 2001). We did not find a lifted version of inequalities (5) for the 2L-CVRP. Finally, constraints (6) impose that all variables are binary. This implies that each vehicle

must serve at least two customers, i.e., as often assumed in the literature single-customer routes are not allowed. Note that this assumption can easily be removed by allowing z_e to assume value 2 when $e \in \delta(0)$.

It is well-known that for the classical CVRP the computation of $r(S)$ is difficult, as it amounts to solving a one-dimensional Bin Packing Problem (1BPP) associated with customer set S and with respect to item weights and vehicle capacity. In our case, this is even more complex since we must also take into account the 2BPP associated with the loading of the items into the vehicles, with the additional side constraint that all the items of any given customer must be loaded into the same bin. Therefore, as it is usually done in the literature, we replace $r(S)$ by a lower bound on its value given by a simple relaxation. In particular, we used

$$r'(S) = \max\left\{\left\lceil \sum_{i \in S} d_i/D \right\rceil, \left\lceil \sum_{i \in S} a_i/A \right\rceil\right\} \tag{7}$$

where the first term is the well-known 1BPP continuous lower bound associated with the customer weights, and the second term is the 2BPP continuous lower bound computed with respect to the total item area of the customers in S. Bound $r'(S)$ may be further strengthened by considering better 2BPP lower bounds, such as those presented in Martello and Vigo (1998), Fekete and Schepers (2001, 2004), and Boschetti and Mingozzi (2003a).

The asymmetric version of the infeasible-path constraints (5) was introduced to formulate the Travelling Salesman Problem with Time Windows in Auscheuer et al. (2000), where also other lifted inequalities can be found. Some of these inequalities were later considered in Bard et al. (2002) to solve the asymmetric version of the CVRP with Time Windows.

Formulation (1)–(6) has the disadvantage of including the large families of constraints (4) and (5). However, by applying a branch-and-cut approach, we actually do not have to include all of them explicitly in the model, but just to define suitable *separation procedures*.

Given a solution z^* of a relaxed model of (1)–(6), the *separation problem* consists of either proving that all constraints in a given family are satisfied by z^*, or finding a violated one. In the next section we introduce simple separation procedures for constraints (4) and (5).

3.2 Separation Procedures

Due to the definition of $r(S)$, constraints (4) induce a separation problem which is $\mathcal{NP}$-Hard (see Naddef and Rinaldi (2002)). Therefore, it is unlikely that we can design a polynomial-time exact algorithm for determining a constraint (4) violated by z^* when it exists. Nevertheless, as it is done for the CVRP, it is possible to propose good heuristic procedures to possibly find such a violated constraint. To this end, we adapted some simple heuristic procedures used for the same constraints in the CVRP (see, e.g., Naddef and Rinaldi (2002)), so as to take into account also the areas of the items. In particular, we used:

Procedure 1: Let us consider the support graph $G^* = (V, E^*)$ of z^* defined by $E^* := \{e \in E : z^*_e > 0\}$ and by a capacity z^*_e associated with edge e. For each customer i, find the min-cut $\delta(S^*)$ separating 0 from i, and such that $i \in S^*$. Then check the constraint (4) defined by $S := S^*$ for potential violation with respect to the relaxed right-hand side $r'(S)$.

Procedure 2: Let us consider a dummy vertex $n+1$ and define a graph $G' = (V \cup \{n+1\}, E')$ where E' contains all the edges in E^* with the same capacities as in Procedure 1, plus a new edge connecting i and $n+1$ with capacity $2d_i/D$. Find the min-cut $\delta(S')$ separating 0 from $n+1$ and such that $n+1 \in S'$, and check constraint (4) defined for $S := S' \setminus \{n+1\}$.

Procedure 3: The same as Procedure 2 but with capacities $2a_i/A$ on the dummy edges.

Let us now address the separation problem associated with infeasible-path constraints (5). Some heuristic separation approaches for inequalities similar to (5) were presented in the literature (see, e.g., Ascheuer et al. (2001) and Bard et al. (2002)). However, for the 2L-CVRP checking the feasibility of a given path is considerably more time consuming and this is the reason why in our approach we separate these constraints only when Procedures 1-3 do not return a violated constraint (4), and when the current solution z^* is integer. It should be observed that in this case z^* defines a solution of 2L-CVRP made up by exactly K routes so that the subsets of customers associated with each different route, as well as the visit order of the customers along the routes, are uniquely determined.

More precisely, we have the following:

Procedure 4: For each route (S, σ) in the current solution, check its feasibility with respect to the loading Conditions 2–4 of Section 3.1, by using the exact algorithm *Check-2L* to be described in Section 4. If route (S, σ) is not feasible, then add the associated constraint (5) to the current model. Otherwise, if all routes are feasible, the overall incumbent solution is possibly updated.

We note that, since the feasibility check of a route is actually an $\mathcal{NP}$-Complete problem, it can happen that procedure *Check-2L* must be stopped whenever a prescribed time limit is exceeded without proving either the feasibility or the infeasibility of the given route.

3.3 Strengthening the LP-relaxation

The extensive literature on the CVRP shows different inequalities that can be added to the linear relaxation of model (1)–(4), see, e.g., Naddef and Rinaldi (2002). An example of these inequalities is represented by the following *multistar inequalities*:

$$\sum_{e \in \delta(S)} z_e \geq \frac{2}{D} \sum_{i \in S} \left(d_i + \sum_{j \in V \setminus (S \cup \{0\})} d_j z_{\{ij\}} \right) \tag{8}$$

for all $S \subseteq V\backslash\{0\}$. These inequalities were introduced by Araque et al. (1990) for the CVRP with unit demands. Later, Fisher (1995) and Gouveia (1995) generalized them to the CVRP with general demands, and Blasum and Hochstättler (2000) and Letchford et al. (2002) provided polynomial separation procedures based on solving max-flow problems. We also note that one could also consider another class of multistar inequalities (8) where item and vehicle loading areas are used, as done with the capacity-cut constraints (see Equation (7)).

3.4 Branching scheme

Whenever the LP-relaxation, strengthened with the additional cuts, does not produce a 2L-CVRP solution, then a branching phase is applied. In our implementation, we used the branch-and-bound framework provided by the CPLEX solver. We conducted an extensive testing on some instances from our benchmark set in order to define the most effective branching scheme and search strategy. We also compared the different search strategies provided by the CPLEX solver and the most effective one was a combination of the "best promising node" and depth-first search (which can be obtained by setting to 0.5 the parameter CPX_PARAM_BTTOOL).

At each node we use the separation procedures described in Section 3.2. In addition, the heuristic algorithm to be described in the next section is applied to the final LP relaxation, both to possibly update the incumbent solution and to detect further violated inequalities of type (5) to be added to the current model.

As previously mentioned, separation procedure 4 requires checking the feasibility of a given route through algorithm *Check-2L*. Since this feasibility check may be very time consuming, we introduced stopping conditions for *Check-2L* based on maximum number of backtrackings and time limit. Whenever one of these conditions is achieved we are clearly neither able to find a feasible loading of the vehicle, nor to prove the infeasibility of the

route. In this case we heuristically consider the route infeasible and add the corresponding (possibly not valid) cut to the current model. We should note that this implies that the final 2L-CVRP solution is not guaranteed to be optimal. However, our extensive computational experience has shown that this is a relatively rare event.

3.5 Heuristic Algorithms

The performance of the overall approach is enhanced by using specific heuristic algorithms for the 2L-CVRP to both derive an initial feasible solution at the root node and to possibly update the incumbent solution at other decision nodes.

A heuristic for the 2L-CVRP should combine the routing component with the two-dimensional loading one. To this end, a set of routes is determined by using a *parametric sequential insertion heuristic* that takes into account both classical routing cost minimization, as well as two-dimensional packing insertion criteria. The obtained routes are improved by means of a two-opt exchange procedure and then checked for feasibility by using procedure *Check-2L* to be defined in Section 4.

The insertion algorithm operates as follows. It constructs one (possibly feasible) route at a time. The route is initialized by connecting the depot to the most distant unrouted customer. The next vertex to be inserted is selected as the one minimizing a modified insertion cost $\widetilde{\Gamma}_i$, that takes into account the simple routing insertion cost, the residual area in the vehicle and the residual weight capacity. More precisely, given a customer i and the current route (S, σ), let us define

$A^{res} = (A - \sum_{i \in S} a_i)$, the residual loading area in the vehicle;

$D^{res} = (D - \sum_{i \in S} d_i)$, the residual weight capacity in the vehicle;

Γ_i , the cost of the best insertion of customer i in the route (S, σ).

The modified insertion cost of customer i in route (S, σ) is then:

$$\widetilde{\Gamma}_i = \alpha\Gamma_i + \beta(A^{res} - a_i) + \gamma(D^{res} - d_i) \tag{9}$$

where α, β, γ are suitably defined non-negative parameters such that $\alpha + \beta + \gamma = 1$. In addition, $\widetilde{\Gamma}_i = \infty$ whenever either $a_i > A^{res}$ or $d_i > D^{res}$. It should be observed that parameters α, β and γ allow the combination of classical routing and packing insertion criteria. For example, when $(\beta = \gamma = 0)$ we have the *cheapest insertion* rule used for the CVRP (see, e.g., Mole and Jameson (1976)). On the other hand, when $\alpha = \gamma = 0$ (or alternatively $\alpha = \beta = 0$) the insertion criterion turns out to be the well-known *first fit* rule used in the bin packing context (see, e.g., Johnson (1973)).

As soon as the current vehicle may not accommodate further customers and the total number of used routes is smaller than K, a new route is initialized and the insertion process is iterated. If all customers were inserted in the K routes, then a first-improvement descent procedure based on a two-opt neighborhood is applied to possibly improve the overall solution cost.

This constructive procedure is repeated ten times with different values of parameters α, β and γ, and the best solution is considered (see Iori (2004) for details). If the resulting solution value is smaller than the actual best known feasible 2L-CVRP solution, then the actual feasibility with respect to the loading of each route is checked through procedure *Check-2L*, to be described in Section 4. The procedure is called with a limit on the CPU time and on the maximum number of backtrackings. If all routes are feasible, then the current best known 2L-CVRP solution is updated. Otherwise, for each route that is proved to be infeasible the corresponding constraint is generated and added to the model.

The above heuristic procedure is executed at the beginning of the branch-and-cut algorithm to possibly produce a first feasible 2L-CVRP solution. In order to exploit the information of the current LP-relaxation at each branch-decision node, this heuristic scheme

is also executed with modified edge costs defined as

$$c'_e = c_e(1 - z^*_e/2) \tag{10}$$

where, for each edge $e \in E$, c_e is the original cost and z^*_e is the value of the associated variable in the current LP solution.

4 A Branch-and-Bound Algorithm for the Loading-Check Problem

The overall solution approach for the 2L-CVRP outlined in the previous section strongly relies on a procedure to check the existence of a feasible loading pattern for a given route (S, σ), according to Conditions 1–4 of Section 2. This problem hereafter is referred to as 2L. It is easy to see that 2L is $\mathcal{NP}$-Complete in the strong sense, since it generalizes other two-dimensional single-bin filling problems, such as the one described in Martello et al. (2000), that are special cases of 2L where $|S| = 1$.

In this section we describe an algorithm, called *Check-2L*, to exactly solve problem 2L. The algorithm is based on a branch-and-bound approach, and extends the one proposed by Martello et al. (2000) for a two-dimensional single-bin filling problem arising within the solution of the three-dimensional Bin Packing Problem (3BPP).

At the root node of the branch-and-bound search tree we compute lower bounds to possibly avoid entering the actual enumeration scheme. To this end, we relax the sequential loading requirements and use the lower bounds proposed by Martello and Vigo (1998) for the 2BPP. If the largest lower bound value is greater than one, then clearly the 2L instance is infeasible. Otherwise, the items are sorted in reversed customer loading sequence (i.e., the first customer to be visited is the last one to be loaded on the vehicle), and by non-increasing width within each customer (breaking ties by non-increasing height).

We then use a simple heuristic procedure, derived from the classical bottom-left ap-

proach (see, e.g., Berkey and Wang (1987)), to possibly detect a feasible single-bin loading pattern. The heuristic places one item at a time (in the given order) in the lowest feasible position in the loading surface and left-justified, checking that Conditions 2–4 are not violated.

Whenever the lower and upper bounds did not prove the feasibility or infeasibility of the route (S, σ), an implicit enumeration scheme is used. The scheme starts with an empty solution and generates one descendant node for each item, in the given order, by placing it in the bottom-left position (i.e., at coordinates $(0, 0)$) of the loading surface.

At each subsequent level of the search tree, a new item is placed in a feasible position of the loading surface. Given a partial solution associated with a node of the search tree, where some items have been placed, descendant nodes are generated, one for each non-placed item and for each feasible placement position.

Scheithauer (1997) showed that the set of feasible placement positions for the 2BPP can be limited to the finite set of the *contour points* of the *contour* associated with the already placed items (similar results were described by Martello et al. (2000) for the 3BPP where such points were called *corner points*). In the 2L case we have to consider additional placement positions as a consequence of the customer sequencing constraints, that prevent the placement of an item at positions that would block the removal of other items belonging to customers visited earlier in the route.

More precisely, given a partial loading associated with a route (S, σ) and the current item to be placed, say (i, ℓ), two different cases may happen. The simplest one arises when item (i, ℓ) does not block any of the already placed items (i', ℓ'), i.e., when $\sigma(i) \leq \sigma(i')$. Then, the only placement positions to be considered for (i, ℓ) are the contour points defined by Scheithauer (1997). Such situation is illustrated through a simple example depicted in Figure 2, in which we consider a vehicle loading surface with $W = 20$ and $H = 30$. The figure gives a partial loading associated with a route (S, σ), with $S = \{1, 2, 3\}$ and

$\sigma(1) = 1$, $\sigma(2) = 2$ and $\sigma(3) = 3$, where four items of these customers have already been placed. The current contour is indicated in the figure by a thick dashed line, and the set of contour points is made up by the two positions $(0, 20)$ and $(7, 16)$, indicated by empty circles. We note that, as explained in Scheithauer (1997), we can consider the area above item $(1, 1)$ and below the contour as "trapped" and not available for placement of items in the descendant nodes, thus allowing for a reduction of the enumeration tree size. If the next item to be placed belongs to customer 1 it will not block the previously placed items, hence its only placement positions are the contour points (0,20) and (7,16).

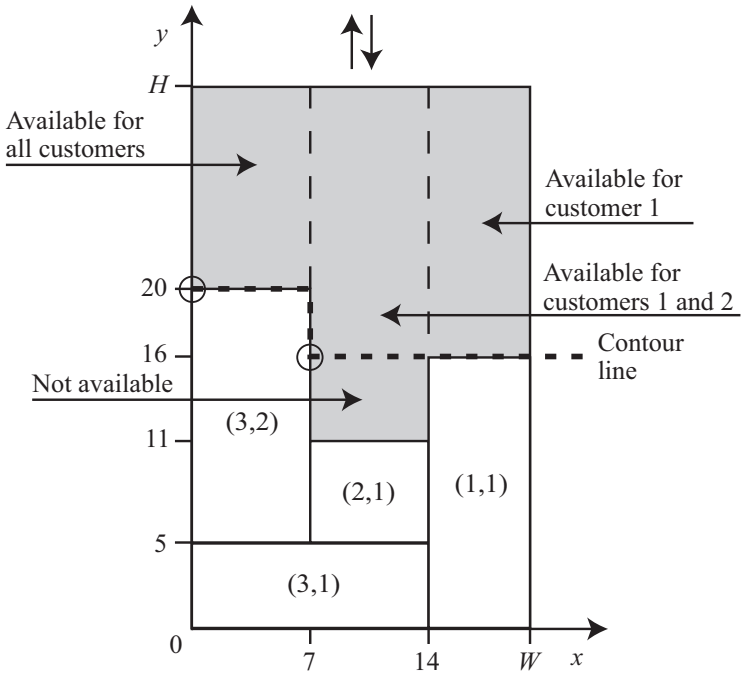

Figure 2: A partial loading with areas available for the next items to be placed. Route (S, σ) has $S = \{1, 2, 3\}$ and $\sigma(1) = 1$, $\sigma(2) = 2$ and $\sigma(3) = 3$.

The second case arises when the current item (i, ℓ) may block already placed items. (In the example of Figure 2 this happens when the next item to be placed belongs to customers 2 or 3.) For the sake of simplicity, let us consider that there is a single already placed item, say (i', ℓ'), that may be blocked by (i, ℓ). This incompatibility removes the area above item

(i', ℓ') from the area where item (i, ℓ) may be placed. Intuitively this splits the area above the contour in at most two new subareas where the placement positions for item (i, ℓ) have to be found. Since in each such subarea blocking incompatibilities are no longer present, the placement positions for (i, ℓ) are the union of those determined within each subarea. It is easy to observe that each item (i', ℓ') that may be blocked by (i, ℓ) can create at most one new placement position with respect to the original set of contour points determined when blocking incompatibilities are ignored. In the example of Figure 2, if the item to be placed belongs to customer 2, the area above item (1,1) is not available, and the resulting additional placement position is (14,16), indicated by a filled circle.

When $p > 1$ items may be blocked by the current item (i, ℓ), we observe that area above the contour is split in at most $p + 1$ different subareas. Thus, in practice, the overall set of placement positions for (i, ℓ) may include at most p additional points with respect to the set of the original contour points.

Whenever an item is placed, we may perform feasibility tests and compute lower bounds to possibly fathom the current node of the enumeration tree. Martello et al. (2000) used for the single-bin 2BPP and 3BPP, a simple bound that estimates the total area, T, that can be placed given a partial allocation, as the sum of the areas of the already placed items, plus the total available area above the contour. If T is not greater than the largest total placed area T^* found so far, then the current node can be clearly fathomed since it may not lead to an improved solution.

In our case, we may define additional simple feasibility tests by again considering the subdivision of the area above the contour imposed by the sequencing constraints. In particular, for each customer we may define the subareas that may be used to place his items. Then, we may fathom the current node as soon as the available area for a given customer is smaller than that of his unplaced items. Figure 2 illustrates this by showing the three subareas in which the area above the contour is divided. In addition, we may

fathom the current node if, for a given customer, the total width of adjacent subareas that may accommodate his items is smaller than the width of his widest unplaced item, or when the maximum height of the available subareas is smaller than the height of his tallest unplaced item.

As previously mentioned, procedure *Check-2L* may consume a large portion of the total time required to solve the overall 2L-CVRP. Moreover, since a given route (S, σ) (or a partial route defined by a customers' sub-sequence) can appear in several different solutions generated by the branch-and-cut algorithm for the 2L-CVRP, we store the routes already tested by procedure *Check-2L* in a pool structure, thus avoiding to solve the same 2L instance twice. In particular, whenever a route is equal to or is contained into a feasible route of the pool, we know that it is feasible. Similarly, we know that a given route is infeasible if it is equal to or contains an infeasible route included in the pool.

5 Computational Results

The algorithm has been coded in C and run on a Pentium IV 3 GHz. The branch-and-cut was embedded in the framework provided by the CPLEX 9.0 solver.

In order to test the performance of the algorithm, we defined a set of test instances by extending to the 2L-CVRP some CVRP instances from the literature (see Reinelt (1991) or Toth and Vigo (2002a) for a definition of these CVRP instances). For each CVRP instance, the corresponding 2L-CVRP instance uses the coordinates and the weights d_i associated with the customers and the weight capacity D of the vehicles. The costs of the arcs are integer values obtained by truncating the value of the Euclidean distances between the corresponding endpoints.

We considered five classes of instances (see Table 1), each corresponding to a different way used to generate the items associated with each customer. The first class corresponds to the original CVRP instance (obtained by assigning to each customer a single item having

both sizes equal to 1, and by setting $W = H = n$). As to the remaining classes, the item and bin sizes are generated in a similar way as in the 2BPP literature. For each class $r = 2, \ldots, 5$, the size of the loading surface of the vehicles is set to $W = 20$ and $H = 40$, and the number of items generated for each customer is a uniformly random integer value in the interval $[1, r]$. For each of these items, a possible shape, namely *Vertical*, *Homogeneous* or *Horizontal*, is selected with equal probability, and the corresponding item sizes are randomly generated in the intervals given in Table 1.

Table 1: Classes used for the generation of the items.

		Vertical		*Homogeneous*		*Horizontal*	
Class	m_i	$h_{i\ell}$	$w_{i\ell}$	$h_{i\ell}$	$w_{i\ell}$	$h_{i\ell}$	$w_{i\ell}$
1	1	1	1	1	1	1	1
2	$[1, 2]$	$\left[\frac{4H}{10}, \frac{9H}{10}\right]$	$\left[\frac{W}{10}, \frac{2W}{10}\right]$	$\left[\frac{2H}{10}, \frac{5H}{10}\right]$	$\left[\frac{2W}{10}, \frac{5W}{10}\right]$	$\left[\frac{H}{10}, \frac{2H}{10}\right]$	$\left[\frac{4W}{10}, \frac{9W}{10}\right]$
3	$[1, 3]$	$\left[\frac{3H}{10}, \frac{8H}{10}\right]$	$\left[\frac{W}{10}, \frac{2W}{10}\right]$	$\left[\frac{2H}{10}, \frac{4H}{10}\right]$	$\left[\frac{2W}{10}, \frac{4W}{10}\right]$	$\left[\frac{H}{10}, \frac{2H}{10}\right]$	$\left[\frac{3W}{10}, \frac{8W}{10}\right]$
4	$[1, 4]$	$\left[\frac{2H}{10}, \frac{7H}{10}\right]$	$\left[\frac{W}{10}, \frac{2W}{10}\right]$	$\left[\frac{H}{10}, \frac{4H}{10}\right]$	$\left[\frac{W}{10}, \frac{4W}{10}\right]$	$\left[\frac{H}{10}, \frac{2H}{10}\right]$	$\left[\frac{2W}{10}, \frac{7W}{10}\right]$
5	$[1, 5]$	$\left[\frac{H}{10}, \frac{6H}{10}\right]$	$\left[\frac{W}{10}, \frac{2W}{10}\right]$	$\left[\frac{H}{10}, \frac{3H}{10}\right]$	$\left[\frac{W}{10}, \frac{3W}{10}\right]$	$\left[\frac{H}{10}, \frac{2H}{10}\right]$	$\left[\frac{W}{10}, \frac{6W}{10}\right]$

It should be observed that in these instances there is no correlation between the weight of the items and their overall area. In our preliminary experiments we also extensively generated correlated instances which, however, turned out to be easier to solve than the uncorrelated ones (see Iori (2004) for more details).

For each instance, the number K of available vehicles is determined as follows. For instances of Class 1, K is set equal to the corresponding value in the original CVRP instance, denoted as K_{CVRP}. For the other classes, we heuristically solve a 2BPP associated with the items, by using an adaptation to 2BPP of the heuristic described in Martello et al. (2000) for the 3BPP. Then K is set to the maximum value between the number of bins in this heuristic solution and K_{CVRP}. It should be noted that, since in the heuristic

2BPP approach clustering and sequential loading constraints are not explicitly taken into account, the resulting K may be smaller that the actual minimum number of vehicles needed to load all the items, thus leading to an infeasible 2L-CVRP instance. However, this situation never occurred in our experiments: the exact algorithm was always able to find a feasible solution using this heuristic K value.

We considered 12 CVRP instances and, for each of them, we generated one instance for each class, obtaining in total 60 instances, with up to 35 customers and 114 items. The original CVRP instances and the complete set of 2L-CVRP instances can be downloaded from `http://www.or.deis.unibo.it/research.html`.

We imposed a CPU time limit of 86400 seconds (i.e., one day) to the overall algorithm. The branch-and-bound algorithm of Section 4 was allowed a maximum time of 7200 seconds (i.e., 2 hours) for each execution of *Check-2L* requested by the Separation Procedure 4 of Section 3.2, and a maximum time of 100 seconds as well as a maximum number of 10000 backtrackings for each execution requested by the heuristic algorithm of Section 3.5. These values were determined during a preliminary tuning of the algorithm, performed by considering a small subset of instances, and represent a good compromise between computing time and overall performance.

The results of our computational experience are summarized in Tables 2 and 3. For each instance, *Name* and *Class* give the name of the original CVRP instance and the class used for the creation of the items, respectively. Then, n represents the number of customers, M the total number of items and K the number of available vehicles.

The performance of the algorithm in the solution of each instance is summarized in three groups of columns denoted as *Loading*, *Routing*, and *Overall*. For what concerns the loading group, *Pool* reports the size of the pool of the feasible and infeasible loading constraints kept in memory, and N_{BB} gives the total number of times algorithm *Check-2L* was called by the Separation Procedure 4 of Section 3.2. $Fails$ gives the number of calls

in which *Check-2L* was not able either to find a feasible loading for the subproblem, or to prove its infeasibility. Finally, T_{load} represents the total CPU time in seconds used by the loading-related procedures. It is worth noting that the pool is filled with N_{BB} (feasible or infeasible) entries by the Separation Procedure 4, and $Pool - N_{BB}$ entries by the heuristic approach of Section 3.5.

In the routing group, $Cuts$ gives the number of cuts obtained through Separation Procedures 1–4 and added to the reduced model (not including default cuts added by CPLEX), $\%_{gap}$ reports the percentage gap between the value z of the best feasible solution found and the final lower bound LB of the root node (computed as $\%_{gap} = 100(z - LB)/LB$), and T_{rout} gives the total time spent by the routing-related procedures. Finally, in the overall group, z gives the value of the best feasible solution found, T_z the time required to obtain such a solution (i.e., the time in which the last update of the incumbent solution was made), and T_{tot} reports the computing time used by the overall algorithm (i.e., $T_{tot} = T_{load} + T_{rout}$). All the times are expressed in seconds on a Pentium IV 3 Ghz. We note that a solution is proved to be optimal if and only if it achieves $Fails = 0$ and $\%_{gap} = 0$. These optimal values are depicted in bold characters.

By observing Tables 2 and 3, we may see that the proposed algorithm was able to solve all instances with up to 25 customers within moderate computing time (always smaller than one hour), whereas a few larger instances were not solved within the overall time limit. The size of the solved instances is consistent with the results reported in the literature for the two separate problems that are combined into the 2L-CVRP. Indeed, a basic branch-and-cut approach for the CVRP using just simple separation procedures for constraints (4) may hardly solve instances with more than 30 customers, whereas state-of-the-art branch-and-bound algorithms for 2BPP (see, e.g., Martello and Vigo (1998), Boschetti and Mingozzi (2003b), Fekete and Schepers (2006), and Pisinger and Sigurd (2006)) may solve loading instances similar to those generated in our work with at most 80–100 items.

The effect of the loading component of the 2L-CVRP is witnessed by the fact that both K and the optimal solution value are in general larger than those of the original CVRP instance. For classes 2, 3, 4 and 5, the average increase in the solution value, with respect to the CVRP solution, is, respectively, 6.8%, 6.1%, 5.9% and 5.6% (computed by considering only the instances with proved optimal values for all classes). The few exceptions to this increase in the routing cost are represented by instances of Class 5, in which the item sizes are relatively small.

Procedure *Check-2L* proved to be particularly efficient for the resolution of the 2L. For the complete set of instances, out of 9216 times in which it was called by the Separation Procedure 4, *Check-2L* failed only 19 times to prove the feasibility or infeasibility of the loading. These exceptions arise for one instance of Class 4 and two instances of Class 5, and this is explained by the fact that these classes are characterized by a large number of items to be loaded into each vehicle. This is a typical feature of very difficult 2BPPs. As a consequence, for each CVRP instance, T_{load} augments from Class 2 to Class 5, and in some cases can absorb a very large part of the computing time. We also note that the final solutions of instances $E030-03g$ Class 4 and Class 5 were not proved to be optimal, due to the failures of *Check-2L* that may result in the addition of possibly not valid cuts (see Section 3.4).

When addressing larger CVRP instances, T_{rout} can increase consistently. This is however not surprising, since it is well known that the separation of constraints (4) does not suffice for the quick solution of CVRPs of these sizes through branch-and-cut. All the instances of Class 1 are solved to optimality. Among the other classes (where the routing does not absorb the entire computing time granted to the algorithm), in just three cases the branch-and-cut approach does not manage to explore the whole tree. The number of cuts added during the execution of the algorithm may be quite large and increases with the size of the instance.

6 Conclusions

In this paper we examined the problem of finding the optimal routes for K vehicles used to serve the demand of n customers, consisting of a set of two-dimensional rectangular items with given sizes and weight. Each vehicle has a weight capacity limit and a rectangular loading surface. The problem, denoted as 2L-CVRP, combines the well-known Capacitated Vehicle Routing Problem and the two-dimensional Bin Packing Problem, and leads to a very complicated overall combinatorial problem.

We presented an integer linear programming model for the 2L-CVRP, containing two families of constraints, used to impose the weight capacity-cut constraints and to forbid infeasible loading patterns. Since both families involve an exponentially-growing number of constraints, we adopted a branch-and-cut approach for the exact solution of the model, using heuristic separation procedures to possibly detect violated constraints.

The proposed approach was successfully tested on instances derived from classical CVRP ones, involving up to 35 customers and more than 100 items. Within 24 hours of computing time, the proposed algorithm was able to solve all the instances of the test bed except five cases. Moreover, all the instances with at most 25 customers were solved in less than one hour of computing time.

Acknowledgements

The authors thank the Associate Editor and two anonymous referees for their comments that greatly improved the quality of the final paper. The first and third authors thank the "Ministero dell'Istruzione, dell'Università e della Ricerca" (MIUR) and the "Consiglio Nazionale delle Ricerche" (CNR), Italy, for the support given to this project. The second author thanks the "Ministerio de Educación y Ciencia" (TIC2003-05982-C05-02), Spain. The computational experiments were executed at the Laboratory of Operations Research

(LabOR) of the University of Bologna.

References

[1] J.R. Araque, L.A. Hall and T.L. Magnanti, "Capacitated trees, capacitated routing and associated polyhedra", Discussion paper 9061, Center for Operations Research and Econometrics, Catholic University of Louvain, 1990.

[2] N. Ascheuer, M. Fischetti and M. Grötschel, "A Polyhedral Study of the Asymmetric Traveling Salesman Problem with Time Windows", *Networks* 36, 69–79 (2000).

[3] N. Ascheuer, M. Fischetti and M. Grötschel, "Solving the Asymmetric Traveling Salesman Problem with Time Windows by Branch-and-Cut", *Mathematical Programming* 90, 475–506 (2001).

[4] J.F. Bard, G. Kontoravdis and G. Yu, "A branch-and-cut procedure for the vehicle routing problem with time windows", *Transportation Science* 36, 250–269 (2002).

[5] J.O. Berkey and P.Y. Wang, "Two dimensional finite bin packing algorithms", *Journal of the Operational Research Society* 38, 423–429 (1987).

[6] U. Blasum and W. Hochstättler, "Application of the branch and cut method to the vehicle routing problem", Working paper zaik2000, University of Koeln, 2000.

[7] A. Bortfeldt and H. Gehring, "A hybrid genetic algorithm for the container loading problem", *European Journal of Operational Research* 131, 143–161 (2000).

[8] M.A. Boschetti and A. Mingozzi, "The two-dimensional finite bin packing problem. Part I: New lower bounds for the oriented case", *4OR* 1(1), 27–42 (2003a).

[9] M.A. Boschetti and A. Mingozzi, "The two-dimensional finite bin packing problem. Part II: New lower and upper bounds", *4OR* 1(2), 135–147 (2003b).

[10] C.S. Chen, S.M. Lee and Q.S. Shen, "An analytical model for the container loading problem", *European Journal of Operational Research* 80, 68–76 (1995).

[11] M. Dell'Amico, S. Martello and D. Vigo. "A lower bound for the non-oriented two-dimensional bin packing problem", *Discrete Applied Mathematics* 118, 13–24 (2002).

[12] S.P. Fekete and J. Schepers, "New classes of fast lower bounds for bin packing problems", *Mathematical Programming*, 91:11–31, 2001.

[13] S.P. Fekete and J. Schepers, "A general framework for bounds for higher-dimensional orthogonal packing problems", *Mathematical Methods of Operations Research*, 60:311–329, 2004.

[14] S.P. Fekete and J. Schepers, "An exact algorithm for higher-dimensional orthogonal packing", to appear in *Operations Research*, (2006).

[15] M.L. Fisher, "Optimal solution of the vehicle routing problems using k-trees", *Operations Research* 42, 621–642 (1995).

[16] R. Fukasawa, H. Longo, J.L. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, and F. Werneck. "Robust branch-and-cut-and-price for the capacitated vehicle routing problem", in *Proceedings of the X IPCO*, volume 3064, pages 1–15, New York, 2004. Springer Lecture Notes in Computer Science.

[17] L. Gouveia, "A result on projection for the vehicle routing problem", *European Journal of Operational Research* 83, 610–624 (1995).

[18] M. Iori, *Metaheuristic algorithms for combinatorial optimization problems*, Ph.D. thesis, DEIS, University of Bologna, Italy (2004).

[19] D.S. Johnson, *Near-optimal bin packing algorithms*, Ph.D. thesis, MIT, Cambridge, MA (1973).

[20] A.N. Letchford, R.W. Eglese and J. Lysgaard, "Multistars, partial multistars and the capacitated vehicle routing problem", *Mathematical Programming* 94(1), 21–40 (2002).

[21] S. Martello, D. Pisinger and D. Vigo, "The three-dimensional bin packing problem", *Operations Research* 48, 256–267 (2000).

[22] S. Martello and D. Vigo, "Exact solution of the two-dimensional finite bin packing problem", *Management Science* 44, 388–399 (1998).

[23] R.H. Mole and S.R. Jameson, "A sequential route-building algorithm employing a generalized savings criterion", *Operational Research Quarterly* 27, 503–511 (1976).

[24] D. Naddef and G. Rinaldi, "Branch-and-cut algorithms for the capacitated VRP", in *The Vehicle Routing Problem*, P. Toth and D. Vigo (eds), 53–84, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002).

[25] D. Pisinger, "A tree search heuristic for the container loading problem", *Ricerca Operativa* 28(87), 31–48 (1998).

[26] D. Pisinger, "Heuristics for the container loading problem", *European Journal of Operational Research* 141, 143–153 (2002).

[27] D. Pisinger, E. Den Boef, J. Korst, S. Martello and D.Vigo. "A note on robot-packable and general variants of the three-dimensional bin packing problem", *Operations Research* 53, 735–736 (2005).

[28] D. Pisinger and M. Sigurd, "Using decomposition techniques and constraint programming for solving the two-dimensional bin packing problem", to appear in *INFORMS Journal on Computing*, (2006).

[29] G. Reinelt, "TSPLIB - a travelling salesman problem library", *ORSA Journal on Computing* 3, 376–384 (1991).

[30] G. Scheithauer, "Algorithms for the container loading problem", in *Operations Research Proceedings 1991*, Springer, Berlin (1992).

[31] G. Scheithauer, "Equivalence and dominance for problems of optimal packing of rectangles", *Ricerca Operativa* 27(83), 3–34 (1997).

[32] G. Scheithauer, "LP - based bounds for the container and multi-container loading problem", *International Transactions in Operational Research* 6, 199–213 (1999).

[33] P. Toth and D. Vigo, "An overview of vehicle routing problems", in *The Vehicle Routing Problem*, P. Toth and D. Vigo (eds), 1–24, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002a).

[34] P. Toth and D. Vigo, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002b).

[35] A. Türkay, "Loading sequence issues in routing problems with packing constraints", Technical report, Uludag Universitesi, Müh. Mim. Fak., Endüstryi Mühendisligi Bölümü, 2003.

[36] H. Xu, Z.-L. Chen, S. Rajagopal and S. Arunapuram, "Solving a practical pickup and delivery problem", *Transportation Science* 37(3), 347–364 (2003).

Table 2: Overall performance of the algorithm.

					Loading				*Routing*			*Overall*		
Name	*Class*	n	M	K	*Pool*	N_{BB}	*Fails*	T_{load}	*Cuts*	$\%_{gap}$	T_{rout}	z	T_{heur}	T_{tot}
$E016-03m$	1	16	15	3	26	9	**0**	0.00	209	**0.00**	0.94	**273**	0.77	0.94
	2	16	24	3	88	40	**0**	5.34	702	**0.00**	10.19	**285**	6.42	15.53
	3	16	31	3	92	29	**0**	16.70	601	**0.00**	4.31	**280**	18.64	21.02
	4	16	37	4	29	9	**0**	3.06	28	**0.00**	0.03	**288**	2.67	3.09
	5	16	45	4	12	5	**0**	40.01	4	**0.00**	0.02	**279**	40.03	40.03
$E016-05m$	1	16	15	5	29	7	**0**	0.00	145	**0.00**	0.36	**329**	0.19	0.36
	2	16	25	5	43	19	**0**	0.06	397	**0.00**	4.28	**342**	3.41	4.34
	3	16	31	5	37	8	**0**	0.27	379	**0.00**	3.59	**347**	0.88	3.86
	4	16	40	5	29	4	**0**	18.94	212	**0.00**	0.92	**336**	0.20	19.86
	5	16	48	5	20	0	**0**	0.00	111	**0.00**	0.27	**329**	0.09	0.27
$E021-04m$	1	21	20	4	44	20	**0**	0.00	352	**0.00**	5.08	**351**	5.05	5.08
	2	21	29	5	1555	706	**0**	9.29	2937	**0.00**	486.71	**396**	467.44	496.00
	3	21	46	5	40	25	**0**	12.96	133	**0.00**	0.90	**387**	13.69	13.86
	4	21	44	5	73	39	**0**	58.38	91	**0.00**	0.31	**374**	57.53	58.69
	5	21	49	5	23	10	**0**	0.20	46	**0.00**	0.06	**369**	0.23	0.27
$E021-06m$	1	21	20	6	40	12	**0**	0.00	93	**0.00**	0.20	**423**	0.17	0.20
	2	21	32	6	34	7	**0**	0.00	238	**0.00**	2.03	**434**	0.19	2.03
	3	21	43	6	85	26	**0**	4.28	587	**0.00**	7.01	**432**	9.70	11.30
	4	21	50	6	53	8	**0**	0.78	368	**0.00**	5.36	**438**	4.92	6.14
	5	21	62	6	38	11	**0**	46.36	154	**0.00**	0.37	**423**	23.59	46.74
$E022-04g$	1	22	21	4	20	8	**0**	0.00	41	**0.00**	0.03	**367**	0.03	0.03
	2	22	31	4	52	27	**0**	1.14	126	**0.00**	1.16	**380**	1.41	2.30
	3	22	37	4	41	10	**0**	1.83	52	**0.00**	0.06	**373**	1.83	1.89
	4	22	41	4	118	31	**0**	32.19	383	**0.00**	3.24	**377**	35.33	35.42
	5	22	57	5	38	18	**0**	1867.22	39	**0.00**	0.16	**389**	1867.25	1867.38
$E022-06m$	1	22	21	6	51	19	**0**	0.02	519	**0.00**	5.77	**488**	5.44	5.78
	2	22	33	6	93	21	**0**	0.39	1010	**0.00**	26.13	**491**	26.11	26.52
	3	22	40	6	77	44	**0**	2.83	1081	**0.00**	34.01	**496**	28.66	36.84
	4	22	57	6	40	0	**0**	1.55	311	**0.00**	1.47	**489**	2.00	3.02
	5	22	56	6	57	29	**0**	0.17	509	**0.00**	4.56	**488**	3.89	4.74

Table 3: Overall performance of the algorithm.

					Loading				*Routing*			*Overall*		
Name	*Class*	n	M	K	*Pool*	N_{BB}	*Fails*	T_{load}	*Cuts*	$\%_{gap}$	T_{rout}	z	T_{heur}	T_{tot}
$E023-03g$	1	23	22	3	3	3	**0**	0.00	30	**0.00**	0.00	**558**	0.00	0.00
	2	23	32	5	407	198	**0**	26.30	787	**0.00**	15.99	**724**	41.27	42.28
	3	23	41	5	44	21	**0**	3.13	47	**0.00**	0.14	**698**	2.28	3.27
	4	23	51	5	93	49	**0**	1958.09	304	**0.00**	1.53	**714**	1959.13	1959.63
	5	23	55	6	64	37	**0**	787.25	20	**0.00**	0.08	**742**	776.53	787.33
$E023-05s$	1	23	22	5	23	10	**0**	0.00	6	**0.00**	0.03	**657**	0.02	0.03
	2	23	29	5	493	321	**0**	1.99	705	**0.00**	16.75	**720**	9.36	18.74
	3	23	42	5	234	91	**0**	54.53	384	**0.00**	6.34	**730**	60.63	60.88
	4	23	48	5	34	10	**0**	18.58	37	**0.00**	0.02	**701**	10.94	18.59
	5	23	52	6	13	6	**0**	958.64	8	**0.00**	0.00	**721**	958.64	958.64
$E026-08m$	1	26	25	8	81	31	**0**	0.02	727	**0.00**	19.78	**609**	18.86	19.80
	2	26	40	8	141	12	**0**	0.02	1199	**0.00**	57.53	**612**	47.00	57.55
	3	26	61	8	139	22	**0**	3.83	1244	**0.00**	160.39	**615**	153.23	164.22
	4	26	63	8	71	18	**0**	8.17	1440	**0.00**	250.03	**626**	158.67	258.20
	5	26	91	8	63	10	**0**	31.91	527	**0.00**	11.27	**609**	18.53	43.17
$E030-03g$	1	30	29	3	73	61	**0**	0.49	6207	**0.00**	54635.13	**524**	25902.42	54635.61
	2	30	43	6	3478	957	**0**	232.08	4546	**0.00**	10035.87	**687**	802.95	10267.95
	3	30	49	6	374	220	**0**	143.58	1068	**0.00**	289.65	**637**	392.92	433.23
	4	30	72	7	1109	270	2	23525.76	1280	**0.00**	174.00	738	23676.27	23699.77
	5	30	86	7	262	94	7	74586.73	427	**0.00**	116.34	706	74231.45	74703.07
$E033-03n$	1	33	32	3	47	32	**0**	0.00	319	**0.00**	6.14	**1991**	6.03	6.14
	2	33	44	7	5943	4555	**0**	342.94	21899	9.29	86058.06	2884	70581.38	86401.01
	3	33	56	7	1074	556	**0**	153.12	2716	**0.00**	14087.63	**2574**	6509.33	14240.75
	4	33	78	7	1588	276	**0**	12232.80	3119	0.15	74167.61	2668	18574.39	86400.41
	5	33	102	8	116	50	10	86398.77	108	4.87	1.25	2739	47714.29	86400.02
$E036-11h$	1	36	35	11	113	27	**0**	0.00	1962	**0.00**	2114.28	**682**	1334.94	2114.28
	2	36	56	11	136	29	**0**	0.02	1857	**0.00**	1426.03	**682**	1333.81	1426.05
	3	36	74	11	162	0	**0**	0.44	1589	**0.00**	992.91	**682**	194.06	993.34
	4	36	93	11	247	49	**0**	120.71	3413	**0.00**	7826.02	**691**	7639.74	7946.74
	5	36	114	11	64	0	**0**	0.12	1637	**0.00**	1139.74	**682**	55.53	1139.86