

A physical packing sequence algorithm for the container loading problem with static mechanical equilibrium conditions

A. Galvão Ramos, José F. Oliveira and Manuel P. Lopes

Abstract

The container loading problem (CLP) is a combinatorial optimization problem for the spatial arrangement of cargo inside containers so as to maximize the usage of space. The algorithms for this problem are of limited practical applicability if real-world constraints are not considered, one of the most important of which is deemed to be stability. This paper addresses static stability, as opposed to dynamic stability, looking at the stability of the cargo during container loading. This paper proposes two algorithms. The first is a static stability algorithm based on static mechanical equilibrium conditions that can be used as a stability evaluation function embedded in CLP algorithms (e.g. constructive heuristics, metaheuristics). The second proposed algorithm is a physical packing sequence algorithm that, given a container loading arrangement, generates the actual sequence by which each box is placed inside the container, considering static stability and loading operation efficiency constraints.

Keywords: physical packing sequence; container loading; 3D packing; static stability

1. Introduction

The transportation sector is currently facing a number of challenges within Europe. The European Commission has cited an efficient transport system as essential to maintaining the EU's prosperity; faced with the present scenario of rising oil prices, growing congestion, and looming climate change, a new strategy for the transport systems in Europe is needed. The strategy established by the EU aims to achieve a 60% reduction in CO₂ emissions and oil dependency by 2050, without limiting freedom of movement. A strong focus is laid on optimizing the performance of multimodal logistic chains, and on increasing the efficiency of transport with information systems (Kallas, 2011).

It is thus expected that in the next decades freight transportation costs will rise, and the need for increased efficiency and security in the transportation sector will drive companies to adopt cargo planning tools that can be effectively used in practice.

It is not then surprising the amount of research that has been carried out focusing on the optimization of the spatial arrangement of cargo inside transportation vehicles or containers.

This problem is known in the literature as the container loading problem (CLP), which belongs to the more generic combinatorial optimization class of Cutting and Packing problems. In the CLP, a set of rectangular shaped boxes (small items) must be packed orthogonally in a set of containers (three dimensional, rectangular, large objects), in a way that the boxes do not overlap and all the boxes of the subset lie entirely within the container. As an assignment problem, there are two possible objectives: output value maximization and input value minimization. The former refers to problems where the number of containers is not sufficient to accommodate all the boxes and therefore the objective is to maximize the value of boxes assigned to all the containers. The latter refers to problems where the number of containers is sufficient to accommodate all the boxes, such that the objective is to minimize the cost of the containers used (Wäscher et al., 2007).

However, the majority of the approaches to the CLP found in the literature are of limited applicability in practical situations. The existence of a series of real-world constraints that strongly influence the loading pattern of a container contributes to that limitation (Bischoff and Ratcliff, 1995; Bortfeldt and Gehring, 2001). One of these constraints is cargo stability, which several authors consider to be the most important.

This paper aims to help reduce the discrepancies between scientific outcomes and real-world needs, firstly, by refining the concept of cargo stability and developing a new static stability evaluation function within the CLP and, secondly, by determining the actual loading sequence of a given packing arrangement, taking into account static stability and loading operations efficiency constraints. The approach followed is based on the assumption that stability should be addressed by separating static stability from dynamic stability, that is, stability of the cargo during loading operations from stability during transportation.

This paper presents a static stability algorithm (SSA) that approaches static stability based on the static mechanical equilibrium conditions applied to rigid bodies derived from Newton's laws of motion, and a physical packing sequence algorithm (PPSA) that evaluates the feasibility of an existing loading arrangement that uses the SSA to evaluate the box placement stability.

The remainder of this paper is organized as follows. Section 2 presents a literature review covering the two main topics of this paper, static stability and the physical packing sequence. In Section 3, an approach to static stability is presented and an algorithm for static stability evaluation is proposed. Section 4 presents the efficiency approach to the loading operations. In Section 5, a PPSA is proposed, followed by the presentation of computational results in Section 6. Finally, Section 7 draws some conclusions from the findings.

2. Literature review

2.1. Static stability

The stability constraint is considered one of the most important CLP constraints. In a recent literature review of container loading constraints, Bortfeldt and Wäscher (2013) note the particular relevance of stability constraints in the literature. Previous works that address cargo stability use the term “stability” in a rather simplified way, and often as if the term was self-explanatory.

Different concepts, such as loading operations stability and transportation stability, are sometimes not addressed separately but dealt with indiscriminately. In the literature, some approaches to stability make a distinction between static (vertical) and dynamic (horizontal) stability, that is, the stability of cargo during loading operations and the stability of cargo during transportation, but the majority of literature approaches only focus on static stability (Bortfeldt and Wäscher, 2013).

Three different types of approaches to static stability can be found in the literature: the full base support, the partial base support and the static mechanical equilibrium approach. Of the three approaches, full base support guarantees static stability, while the proposed approaches for partial base support and static mechanical equilibrium do not guarantee static stability.

The full base support approach requires the entire base of a box to be in contact with the base of the container or with the top surface of other boxes. As a result no overhanging boxes are allowed. Examples can be found in Bischoff and Ratcliff (1995), Bortfeldt and Gehring (2001), Gonçalves and Resende (2012), and Zhu and Lim (2012). This approach excessively penalizes space usage in the container (Bischoff and Ratcliff, 1995), and does not necessarily meet real-world needs.

In the case of partial base support, either the entire base of a box is required to be in contact with the base of the container, or a pre-specified percentage of the area of a box base must be in contact with the top surface of other boxes. As an example, Carpenter and Dowsland (1985) require the contact area to fall in the range 95–75%, while Christensen and Rousøe (2009) require a minimum of 80%, Gendreau et al. (2006), Fuellerer et al. (2010), and Tarantilis et al. (2009) 75%, Gehring and Bortfeldt (1997) 70%, and Mack et al. (2004) 55%.

In the static mechanical equilibrium approach, requirements are as follows:

- The entire base of a box be in contact with the base of the container or,
- the center of gravity of a box be located above the contact surface of the supporting boxes.

Even though the center of gravity condition is not referred to in the literature as a static mechanical equilibrium condition (Lin et al., 2006; Mack et al., 2004), it does in fact derive from static mechanical equilibrium conditions applied to rigid bodies. The first authors to actually mention the conditions within the CLP were de Castro Silva et al. (2003). Their approach to static stability is divided into two parts. In the first part it is checked if at least one of the following three conditions is met: (1) the base of the box is fully in contact with the base of the container; (2) the center of gravity of the box is directly above a supporting box; (3) the center of gravity of the box is not directly above a supporting box, but there is a vertical plane passing through the center of gravity and connecting two nonadjacent supporting boxes. The second part addresses the resultant moments for a set of boxes, stating that if the sum of the moments for every box of the set is null then the set is considered to be stable. The approach presented by de Castro Silva et al. (2003) has two practical limitations: no algorithm or method is proposed for checking condition 3 (the existence of a vertical plane) or for evaluating the sum of the moments.

Another aspect of recent approaches in the literature is the replacement of static stability constraints by the full base support constraint (Gonçalves and Resende, 2012; Zhu and Lim, 2012). This is particularly clear in the performance evaluation of the proposed solutions. The metrics used for performance benchmarking algorithms are usually the percentage of volume loaded with or without full base support, and consequently the solutions are obtained by imposing full base

support, thus assuring stability. As a result, the goal is not to obtain a loading arrangement that is statically stable but a loading arrangement where all boxes have full base support.

2.2. Physical packing sequence

Evaluating the stability of a cargo arrangement during loading operations is intimately related with the actual packing sequence of boxes inside the container. The physical packing sequence is the sequence by which each box is placed inside the container in a specific location determined by the CLP algorithm. This sequence is unlikely to correspond to the generation order of the positions and spatial orientation for the boxes in the cargo arrangement (Bischoff and Ratcliff, 1995; Ngoi et al., 1994).

The generation of a cargo loading sequence to be used in practice is not taken into account by the majority of existing CLP algorithms that focus on space optimization. This issue is only tackled in situations where the loading or unloading sequence is directly related with the problem characteristics (e.g. multidrop situations), or is directly related with the nature of the problem, as when the CLP is combined with vehicle routing problem (VRP) (Bortfeldt and Wäscher, 2013).

Even though the physical packing sequence is usually ignored in the existing CLP algorithms, some authors mention the possibility of using an algorithm that is independent of that in the CLP to generate the packing sequence. However, no such algorithm is typically provided (Bischoff and Ratcliff, 1995; Bortfeldt et al., 2003; Ngoi et al., 1994).

The physical packing sequence is also dependent on the presence of loading or unloading constraints and is usually addressed together with the ease of the loading operation. The ease of the loading operation reflects the effort required to handle the boxes during loading.

The ease of loading or unloading is also usually addressed in problems where the CLP and VRP are combined (Fuellerer et al., 2010; Gendreau et al., 2006; Iori and Martello, 2010; Moura and Oliveira, 2008; Tarantilis et al., 2009) or in situations of cargo multidropping (Ceschia and Schaerf, 2011; Christensen and Rousøe, 2009; Junqueira et al., 2011, 2013). In both situations, boxes have to be delivered to different destinations (customers) and must be located close to each other inside the container. The goal is to have all the boxes of a destination unloaded without moving any of the boxes of each of the subsequent destinations. This would make unloading easier and avoid unnecessary unloading and reloading operations (Fuellerer et al., 2010; Moura and Oliveira, 2008).

The loading arrangement, when different destinations are considered, must take into account the sequence by which each customer is served and, therefore, the sequence by which the boxes are unloaded. Usually a *last in first out* (LIFO) strategy is followed so that boxes can be unloaded without the need to rearrange other boxes. This means that in order to unload a box, no other box can be placed either in front or above it. This geometric condition is treated in the CLP as a hard constraint (Christensen and Rousøe, 2009).

Besides the LIFO constraint, other constraints are also considered by some authors when addressing the feasibility of solutions to the CLP under multidrop situations. The maximum reach is a parameter used to show the maximum number of units of length by which the worker can exceed the previous customer's virtual plane, in order to arrange the boxes of the next customer. The virtual plane of a customer is a plane parallel to the front of the container, defined by the customer box surface closest to the container entrance. This parameter, δ_k , illustrated in Fig. 1, can also represent

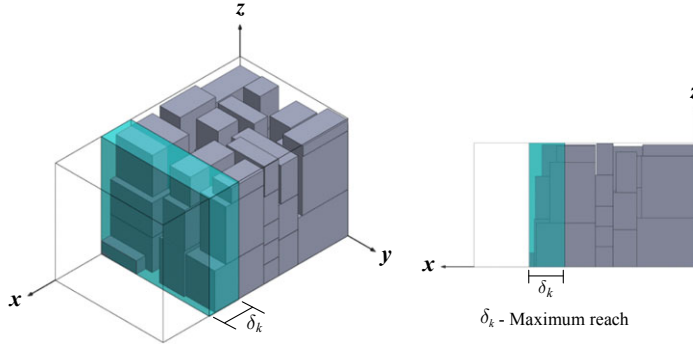


Fig. 1. Maximum reach parameter.

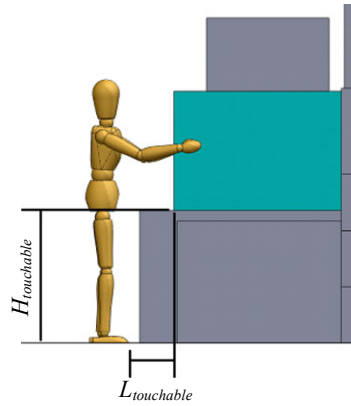


Fig. 2. Untouchable box parameters.

the reach of the worker's arm or even a device used to load/unload the boxes (Junqueira et al., 2011, 2013).

In the context of the home delivery model, which can be considered an extreme case of the multidrop situation, Liu et al. (2011) discuss the unloading cost. The unloading cost measures the effort involved in unloading a box from a container. The authors define the unloading cost of a box as being directly proportional to the number of boxes that have not yet been delivered and that need to be unloaded and reloaded in order to unload the box.

To calculate the unloading cost, Liu et al. (2011) use the invisible and untouchable rule. According to this rule, a box that has not been delivered counts as a cost to another box if it is positioned in front of the box or makes the box untouchable by the worker. To evaluate the latter condition, two variables were defined, touchable length $L_{touchable}$ and touchable height $H_{touchable}$, which are the distances from the box to the feet of the worker along the x and z -axis, respectively (see Fig. 2).

According to these authors, if a box can be touched by the worker the variables should satisfy two constraints:

$$L_{touchable} + H_{touchable} \leq (body\ height + arm\ length)$$

$$L_{touchable} \leq \min\{(body\ height + arm\ length - z_i), arm\ length\},$$

where z_i is the z -axis coordinate origin of box b_i .

3. Static stability

Different strategies have been presented in the literature to address the static stability constraint. The majority of authors require either full base support or almost full base support for the boxes as a condition to guarantee static stability. Considering that these conditions can be very restrictive during the loading of boxes into the container, an approach to stability is presented here based on the static mechanical equilibrium conditions applied to rigid bodies.

There are two conditions of equilibrium for rigid bodies that are derived from Newton's first and third laws of motion. The first law states that "Every object persists in its state of rest or uniform motion in a straight line unless it is compelled to change that state by forces impressed on it." The third law states that "For every action there is an equal and opposite reaction." In other words, if object A exerts a force on object B , then object B also exerts an equal and opposite force on object A (Hibbeler, 2010).

The first condition of equilibrium is

$$\sum \vec{F} = \vec{0}, \quad (1)$$

where \vec{F} represents the external forces applied on a rigid body. A force is a vector quantity that characterizes the force magnitude and the direction of its action (Hibbeler, 2010).

The second condition of equilibrium is

$$\sum \vec{M}_O = \sum (\vec{r} \times \vec{F}) = 0, \quad (2)$$

where \vec{r} represents the vector from point O to the line of action of force \vec{F} , \vec{M}_O represents the moment of a force, that is, the tendency of a force to rotate a body about a point or axis. The first condition provides the translational equilibrium and the second condition the rotational equilibrium of a body (Hibbeler, 2010).

Following Newton's laws, the mass of a system can be treated as a point mass. In the proposed approach, boxes are considered bodies of homogeneously distributed mass. When the body is homogeneous the center of gravity of the body coincides with the geometric center of the body. The center of gravity is the point where an object's weight distribution is considered to be applied (Hibbeler, 2010).

When evaluating the static stability of a cargo arrangement we have to deal with a box arrangement of several items that can move independently of each other, that is, there are several rigid bodies that all have to be in a stable position.

For each box, we have the static stability criteria (1) and (2). Let b be a box placed at position (x, y, z) with depth (d) width (w), and height (h), whose support polygon is denoted by S . The support polygon of a box is formed by the convex hull of all points in contact with the base of

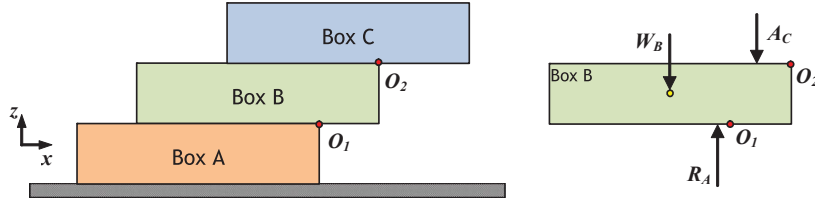


Fig. 3. Free body diagram of box B.

the box. The support polygon concept is frequently used in the research field of human movement simulation for stability modeling purposes (Badler et al., 1980; Vukobratović and Borovac, 2004).

Given that the only external force acting on the set of boxes is the gravitational force, and all the faces of the boxes are parallel or perpendicular to the direction of the gravitational force, when taken in conjunction with (1) implies that all the forces acting on each of the boxes are parallel to the gravitational force.

Box b is subject to the gravitational force (\vec{W}), the action forces (\vec{A}_k) of supported boxes k , and the reaction forces (\vec{R}_j) of support boxes j . The reaction forces are acting on the area of the base of the box in contact with other boxes. Let $\vec{A} = \sum_{k=1}^K \vec{A}_k$ applied at point M so that $\vec{A} \times \vec{OM} = \sum_{k=1}^K (\vec{A}_k \times \vec{OM}_k)$, and $\vec{R} = \sum_{j=1}^J \vec{R}_j$ applied at point Q so that $\vec{R} \times \vec{OQ} = \sum_{j=1}^J (\vec{R}_j \times \vec{OQ}_j)$. Let point O be the origin of the coordinate system. Figure 3 illustrates three stacked boxes A , B , and C , and the free body diagram of box B , which depicts all the relevant forces acting on the box.

Let \vec{F}_R be the resultant force acting downwards parallel to the z -axis on box b , acting on point P with coordinates (x_p, y_p, z_p) . The magnitude of the resultant force \vec{F}_R is established by adding \vec{W} and \vec{A} , that is,

$$\vec{F}_R = \vec{W} + \vec{A}.$$

Application point P is determined by calculating the moment of \vec{F}_R about point O , and the resulting moment of \vec{W} and \vec{A} . The moment of \vec{F}_R about point O equals the sum of moments of the forces \vec{W} and \vec{A} about O , that is,

$$\begin{aligned} \vec{M}_{F_R} &= \vec{M}_W + \vec{M}_A \\ \vec{F}_R \times \vec{OP} &= \vec{W} \times \vec{OG} + \vec{A} \times \vec{OM}, \end{aligned}$$

where \vec{OP} , \vec{OG} , and \vec{OM} are radius vectors from the origin of the coordinate system O to the point where the resultant force acts (P), center of gravity of box b (G), and the points where the action forces are applied (M), respectively.

Proposition 1. *Stability criteria (1) and (2) are met for a box b , subject to the force \vec{F}_R , if the projection of the application point P with coordinates (x_p, y_p, z_p) in plane $(0, 0, z_b)$, lies inside the support polygon S of box b .*

Proof. Applying Equations (1) and (2)

$$\begin{aligned}\sum \vec{F} &= \vec{F}_R + \vec{R} = \vec{0} \\ \sum \vec{M}_O &= \vec{F}_R \times \vec{OP} + \vec{R} \times \vec{OQ} = \vec{0}\end{aligned}$$

reveals that coordinates (x_p, y_p) are the same as coordinates (x_q, y_q) of point Q .

Since the application point of the resultant force of two parallel forces with the same direction and with application points a and b in the same plane perpendicular to the forces is located in a line segment that connects a and b such that $Q = \lambda a + (1 - \lambda)b$ for $0 \leq \lambda \leq 1$, the resultant of the reaction forces acting on the base of a box is always located inside a convex hull defined by all the horizontal support points of the box.

It can then be concluded that if P is inside the support polygon S , then conditions (1) and (2) are satisfied. \square

3.1. Static stability algorithm

The exact task of the SSA is as follows: given a stable arrangement A' of n boxes, the SSA has to decide whether the extended arrangement A consisting of A' and one additional placement (b, x, y, z, d, w, h) is also stable. Hence, we have to answer the question of which placements of A might become unstable by the new placement of b . If the set of these placements $A_r \subseteq A$ is determined, then the task of the SSA is to perform a stability check for the new placement of b and for all placements in A_r .

A placement of a box b consists of a reference point with coordinates (x, y, z) and depth (d), width (w), and height (h), dimensions measured along the x -, y -, and z -axis of a Cartesian coordinates system, after being positioned in the container. The container is considered to be placed in the first octant of the coordinates system with the back-bottom-left corner lying in its origin.

Proposition 2. *The set of possibly affected placements A_r consists of all placements of boxes that support box b and support the supporting boxes of b etc.*

Proof. The proof is clear, since the weight transmission can only take place from b to the boxes in A_r . \square

In the proposed SSA, a box b located at (x_b, y_b, z_b) subject to the force F_R , is considered stable, if the projection of the application point P with coordinates (x_p, y_p, z_p) in plane $(0, 0, z_b)$, lies inside the box b support polygon S . This condition requires that each time an attempt is made to place a box inside the container, the static stability of the subset of boxes A_r loaded in the container must be evaluated, in order to check if the new subset A is stable. Therefore, to guarantee that the static stability condition holds, for all boxes of A_r , one of the three following conditions must be satisfied:

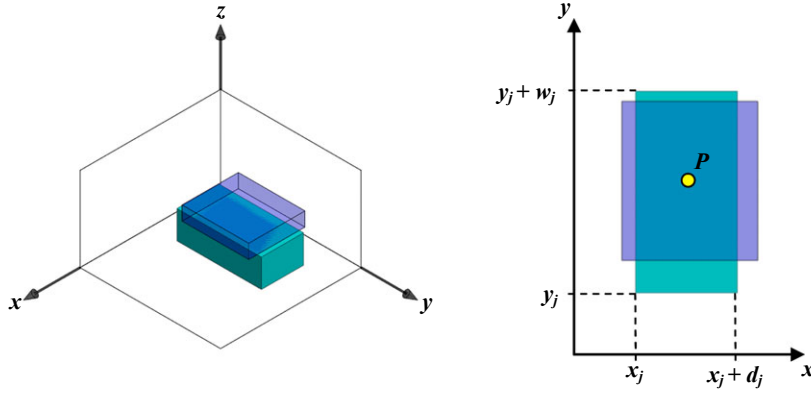


Fig. 4. Example support polygon for condition 2.

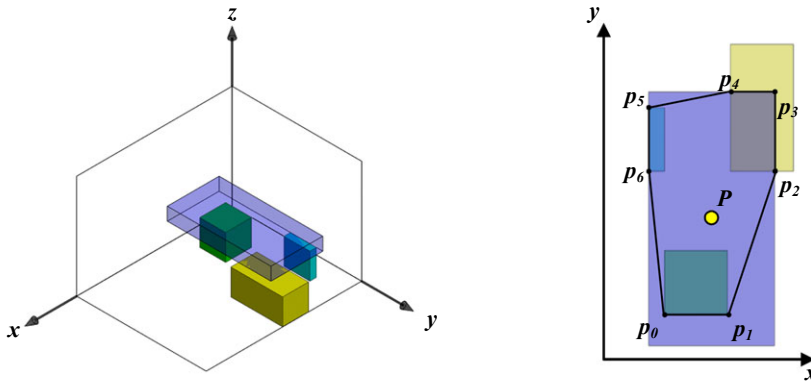


Fig. 5. Example support polygon for condition 3.

1. the support polygon of box b is the container floor, that is, whenever z_b , the z -axis coordinate of a box, equals zero;

$$z_b = 0,$$

2. point P lies in the interior or frontier of a support polygon, defined by the top edges of a support box b_j (Fig. 4). The condition holds for box b , supported by box b_j located at (x_j, y_j, z_j) and with dimensions (d_j, w_j, h_j) , when

$$x_j \leq x_p \leq x_j + d_j$$

$$y_j \leq y_p \leq y_j + w_j$$

$$z_b = z_j + h_j$$

3. point P lies in the interior or frontier of a support polygon S , a convex polygon defined by the convex hull of the vertices of the polygons generated by the intersection of box b with its supporting boxes (Fig. 5). Consider S to be a convex polygon in plane $(0, 0, z_b)$ with m vertices

$(p_0, p_1, \dots, p_{m-1})$ defined by a sequence of points $(x_0, y_0), (x_1, y_1), \dots, (x_{m-1}, y_{m-1})$ given in counterclockwise direction. Condition 3 holds whenever point P is an interior point of support polygon S . Point P , is a point or an interior point of support polygon S , if P is located on a line segment that connects a pair of points a, b in S , such that $P = \lambda a + (1 - \lambda)b$ for $0 \leq \lambda \leq 1$ (LaValle, 2006).

The SSA is described in Algorithm 1.

Algorithm 1. Static_Stability (box b)

Input: Let box b be a box placed at coordinates (x_b, y_b, z_b) with depth (d_b) width (w_b) and height (h_b).

Output: Let *stable* be a boolean variable where TRUE represents a stable box and FALSE an unstable box.

Begin

stable \leftarrow FALSE

if $z_b = 0$ **then**

stable \leftarrow TRUE

return *stable*

end if

Let U be the set of loaded boxes b_j that support box b

Let F_r be the sum of downward forces acting on box b

Let P be the acting point of F_r with coordinates x_p, y_p

for each box $b_j \in U$ **do**

if $(x_j \leq x_p \leq x_j + d_j)$ and $(y_j \leq y_p \leq y_j + w_j)$ **then**

stable \leftarrow TRUE

end if

Determine the intersection vertices v with box b

end for

if *stable* = FALSE **then**

$S \leftarrow$ **Call** Gift wrapping (v)

stable \leftarrow **Call** Point-in-Polygon (P, S)

▷Determine box b support polygon

▷Determine box stability

end if

if *stable* = TRUE **then**

Determine box b reaction forces R_j

for each box $b_j \in U$ **do**

if **Call** Static_Stability(b_j) = FALSE **then**

stable = FALSE

end if

end for

▷Determine system stability

end if

return *stable*

End

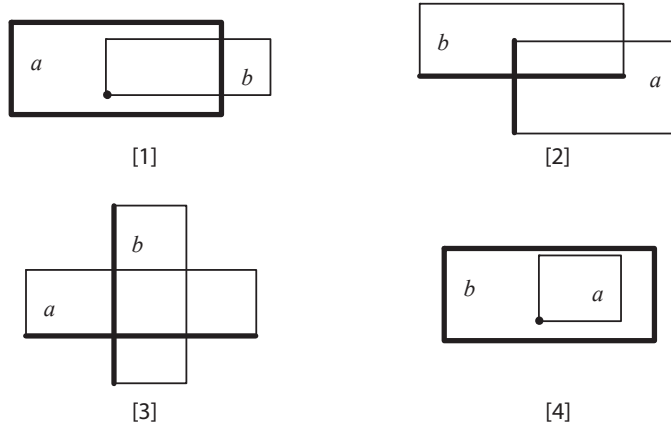


Fig. 6. The four possible types of intersection (adapted from Edelsbrunner and Maurer, 1981).

3.1.1. Construction of the support polygon

In order to be able to evaluate stability condition 3, it is necessary to build the convex polygon induced by the support boxes. The support polygon S represented by vertices $(p_0, p_1, \dots, p_{m-1})$ is the convex hull of the set of points T , which represents the vertices $v_{j,k} = (x_{j,k}, y_{j,k})$ for $k = (1, 2, \dots, 4)$ of the intersection of box b with its support boxes b_j for $j = (1, 2, \dots, n)$. The intersection of a box b with its support box b_j has the shape of a rectangle, since boxes are placed orthogonally to each other.

There are four types of intersection of two-dimensional orthogonal objects. Let a and b be two orthogonal rectangles. Rectangle a intercepts b if at least one of the following conditions, illustrated in Fig. 6, stand (Edelsbrunner and Maurer, 1981):

1. a contains the left bottom point of b ;
2. the left border line of a intercepts the bottom border line of b ;
3. the bottom border line of a intercepts the left border line of b ;
4. the left bottom point of a is contained in b .

Based on the four types of intersection and considering two orthogonal rectangles b and b_j , each defined by the lower left and upper right corner with coordinates $(x_b, y_b), (x_b + d_b, y_b + w_b))$ and $((x_j, y_j), (x_j + d_j, y_j + w_j))$, the vertices of the intersection area can be determined by the following expressions:

$$\begin{aligned}
 v_{j,1} &= (\max(x_b, x_j), \max(y_b, y_j)) \\
 v_{j,2} &= (\min(x_b + d_b, x_j + d_j), \max(y_b, y_j)) \\
 v_{j,3} &= (\max(x_b, x_j), \min(y_b + w_b, y_j + w_j)) \\
 v_{j,4} &= (\min(x_b + d_b, x_j + d_j), \min(y_b + w_b, y_j + w_j)).
 \end{aligned}$$

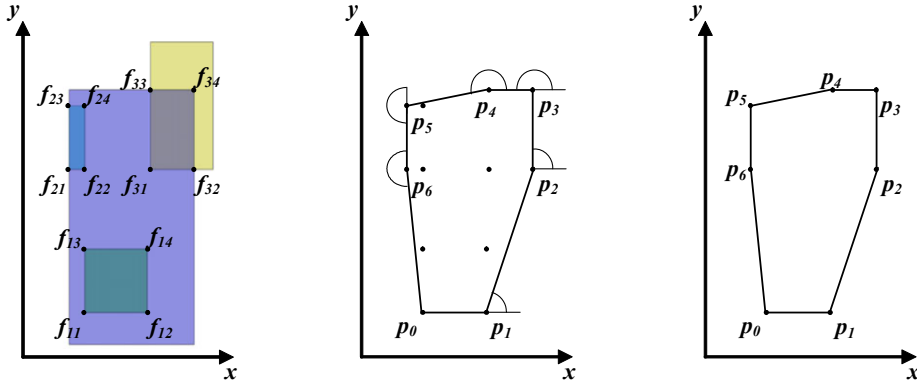


Fig. 7. “Gift Wrapping” algorithm illustration.

The algorithm that generates the convex hull of T is based on the concept of “Gift Wrapping,” which basically simulates the wrapping of a taut piece of paper around a gift (Chand and Kapur, 1970; Dibakar and Mruthyunjaya, 1999; Jarvis, 1973).

It starts by computing the point p_0 with the lowest $y_{j,k}$, since it is known that the most inferior point must be a convex hull vertex. The algorithm proceeds, finding the next convex hull vertex with the smallest counterclockwise angle that is greater than zero with respect to p_0 . Once the successive point in the convex hull is identified, the process continues based on angles with respect to the new point until the next vertex is the initial point. The algorithm also eliminates collinear points. Figure 7 illustrates the algorithm.

Algorithm 2. Gift Wrapping (points)

Input: Let [points] be a vector of *NumPoints* points with coordinates $x_i, y_i, i = 1, \dots, \text{NumPoints}$.

Let $used_i$ be a flag that indicates if a point was added to the convex hull.

Output: Let S be the convex hull of [points].

Begin

for $j = 1$ **to** *NumPoints* **do**

 [points][j].*used* \leftarrow FALSE

end for

Let *minPoint* be the point with the minimum y -axis coordinate (if there is more than one *minPoint*, select the one with the minimum x -axis coordinate)

Call addConvexHullPoint(*minPoint*)

▷add point to convex hull S

points[*minPoint*].*used* \leftarrow TRUE

currPoint \leftarrow *minPoint*

▷set the current point

cont1 \leftarrow 0

repeat

$p1 \leftarrow$ points[*currPoint*]

cont \leftarrow 0

for $k = 1$ **to** *NumPoints* **do**

```

if  $points[k].used = FALSE$  then
  if  $cont = 0$  then
     $p2 \leftarrow points[k]$ 
     $minAngle \leftarrow k$ 
     $cont ++$ 
  else if  $findAngle(p1, points[k]) \leq findAngle(p1, p2)$  then ▷find minimum angle
    if  $findAngle(p1, points[k]) = findAngle(p1, p2)$  then ▷check for collinearity
      if  $distance(p1, points[k]) \leq distance(p1, p2)$  then
         $points[k].used \leftarrow TRUE$ 
        step
      else
         $points[minAngle].used \leftarrow TRUE$ 
      end if
    end if
     $p2 \leftarrow points[k]$ 
     $minAngle \leftarrow k$ 
  end if
end if
end for
if  $cont1 = 0$  then
   $cont1 ++$ 
   $points[minPoint].used \leftarrow FALSE$ 
end if
 $points[minAngle].used \leftarrow TRUE$ 
 $currPoint \leftarrow minAngle$ 
Call  $addConvexHullPoint(currPoint)$ 
until  $currPoint = minPoint$ 
End

```

The pseudocode of the “Gift Wrapping” algorithm is described in Algorithm 2. The function *findAngle* in the algorithm returns the angle between the horizontal line and the line joining two points.

3.1.2. Determine if the application point is on the boundary or inside the support polygon

Following the construction of the support polygon it is necessary to verify if the “application point” is placed on the boundary or inside the polygon. Testing whether a point is inside or on the boundary of a polygon is known in computational geometry as the point-in-polygon problem. The strategy followed to solve the problem is to test the point against each exterior edge in sequence. If the point is outside any edge, then the point must be outside the convex polygon (Haines, 1994; Shimrat, 1962).

The algorithm uses the properties of the vector product between two vectors in order to establish if the point is to the left, to the right, or on the edge. It starts by generating two vectors. One is from the first point (p_i) to the final point (p_{i+1}) of the segment, and the other is from the first point (p_i) of the segment to the application point (p_P). It then calculates the result of the vector product for the vectors generated, and conclusions are drawn from there, as shown in Fig. 8. In this case, $\overrightarrow{p_i p_{i+1}} \times \overrightarrow{p_i p_P}$ is positive if p_P is to the right of the segment, it is negative if p_P is to the left, or

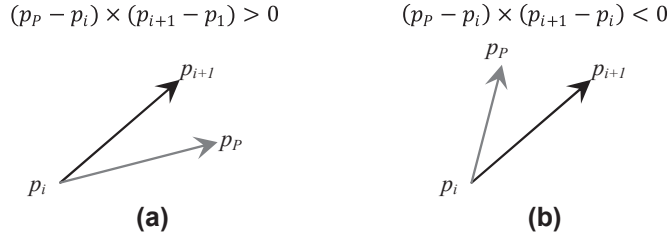


Fig. 8. Point-in-polygon determination.

it is 0 if it is on the segment. Since the polygon was constructed in a counterclockwise direction, it is considered that p_p is inside the polygon if the vector product is negative for all segments of the polygon. The pseudocode of the point-in-polygon algorithm is described in Algorithm 3. The algorithm is an adaptation of the one proposed by Nordbeck and Rystedt (1967).

Algorithm 3. Point-in-Polygon (Point P , Convex Hull S)

Input: Let P , a point with coordinates x_p, y_p , be the application point of force F_R on box b .

Let S be a vector of n points with coordinates $x_i, y_i (i = 1, \dots, n)$, representing the support polygon of box b .

Output: Let *stable* be a boolean value where FALSE represents a point outside the polygon S and TRUE the opposite case.

Begin

stable \leftarrow TRUE

for $i = 1$ **to** $n - 1$ **do**

vectorproduct $\leftarrow (x_p - x_i) * (y_{i+1} - y_i) - (x_{i+1} - x_i) * (y_p - y_i)$

if *vectorproduct* > 0 **then**

stable = FALSE

return *stable*

end if

end for

return *stable*

End

3.1.3. Determine reaction forces

The determination of the reaction forces of all the support boxes of box b starts by writing Equations (1) and (2) as three scalar equations (3). Only three scalar equations are used since there are no forces with a component in both the x - and y -axis, and as consequence no moments about the z -axis are generated.

$$\begin{aligned}
\sum F_z &= 0 \\
\sum M_x &= 0 \\
\sum M_y &= 0
\end{aligned} \tag{3}$$

When the box b equation system is statically indeterminate, that is, if the number of unknowns in (3) is greater than three, additional equations obtained from geometry deformation conditions are added. For this purpose box b is modeled as a rigid body while the support boxes are modeled as deformable bodies with the same linear elasticity. By considering the box b a rigid body, the base of the box forms a plane in space with Equation (4) and is used as a compatibility condition between reaction forces.

$$z = \alpha x + \beta y + z_0 \tag{4}$$

The reaction forces of the support box are then treated as springs with equal stiffness. By choosing one of the support boxes as the origin (z_0), a displacement relation can be established with each of the remaining support boxes, thereby obtaining the compatibility conditions required to solve the problem. This method is used in the field of mechanics of materials and is referred to as the “Force Method of Analysis” (Hibbeler, 1994).

4. Loading/unloading operations efficiency

When defining a cargo arrangement, a high utilization of the container space is important for achieving high efficiency in the transportation system. However, the expected increase in cargo transportation in coming years will also require that other elements of the supply chain, such as warehouses and distribution centers, improve their performance in order to avoid bottlenecks in the supply chain, since the cargo throughput of those points of the supply chain is also expected to increase.

Reducing container cargo loading and unloading time will contribute to reduced bottlenecks in warehouse inbound and outbound operations. (Un)loading operations are usually measured by the number of (un)loaded boxes per man-hour. With effective (un)loading operations, the consequent increase in the number of boxes per man-hour will impact the performance of the supply chain. It is then desirable that the cargo arrangement in the container is influenced by the effectiveness of the (un)loading operations.

There are several techniques for (un)loading cargo from containers or trailers:

- Manual handling—manual (un)loading of cargo done by workers by lifting and carrying boxes to/from pallets or a flexible conveyor system.
- Using handling equipment—(un)loading by means of forklift trucks, pallet trucks or other mechanical handling equipment.
- Automated systems—handling systems that (un)load cargo from/to containers, trucks, or trailers automatically. This can be achieved by using conveyors, rollers, skates, or robot arms.

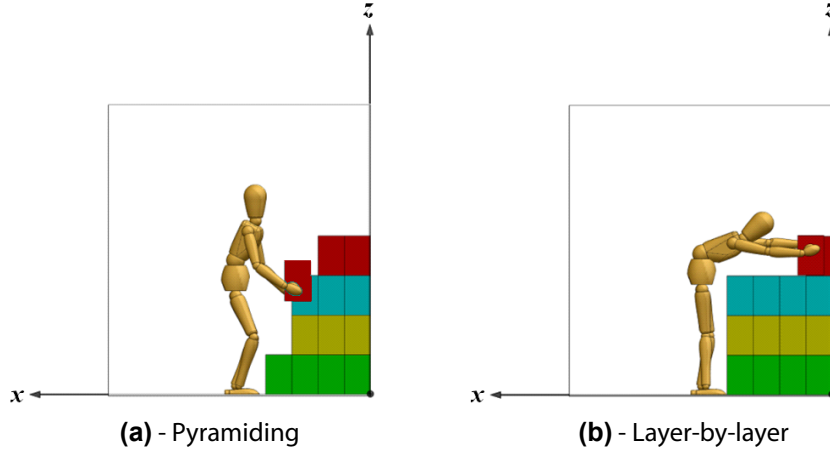


Fig. 9. Stacking methods of container loading.

From an ergonomic perspective, manual handling activities can be physically very challenging and have higher probability of injury to the workers. This is mainly due to muscular fatigue, repetitious movements such as bending, lifting or twisting, carrying loads for a distance, and overhead reaching and lifting (Bhattacharya and McGlothlin, 2012). There are several types of equipment designed to help in lifting boxes which can mitigate the stress that operators are subjected to during manual (un)loading operations.

As with manual loading, these systems require that there is no obstruction when loading, either vertical, that is, along the z -axis, or horizontal, parallel to the x -axis. When a new box b is placed inside container C , a check is made for any obstruction by detecting whether the empty space required to load the box has not been partially filled by any of the boxes already placed. A box b located at (x_b, y_b, z_b) and with dimensions (d_b, w_b, h_b) can be inserted in container C , loaded with boxes b_j , if condition (5) is verified. It considers that if all placed boxes, b_j , are located behind, to the right, to the left, or underneath the final position for box b then definitively there are no obstructions to placing box b .

$$(x_j + d_j \leq x_b) \vee (y_b + w_b \leq y_j) \vee (y_j + w_j \leq y_b) \vee (z_j + h_j \leq z_b). \quad (5)$$

Condition (6) must also be satisfied for those boxes not yet loaded. This condition guarantees that a box b can be loaded if all boxes b_j that are located under box b are already in place, that is, all the remaining boxes b_i to be loaded are to be located behind, in front, to the right, to the left, or above box b .

$$(x_i + d_i \leq x_b) \vee (x_b + d_b \leq x_i) \vee (y_b + w_b \leq y_i) \vee (y_i + w_i \leq y_b) \vee (z_i \geq z_b + h_b). \quad (6)$$

Another relevant aspect to the (un)loading operations is the method by which the boxes are stacked in the container. The two most common loading methods are the layer-by-layer and the pyramiding methods. The pyramiding method consists of (un)loading boxes in a diagonal pattern from the back-bottom to front-top. In the layer-by-layer method boxes are (un)loaded one layer at a time within reachable arm's length (see Fig. 9).

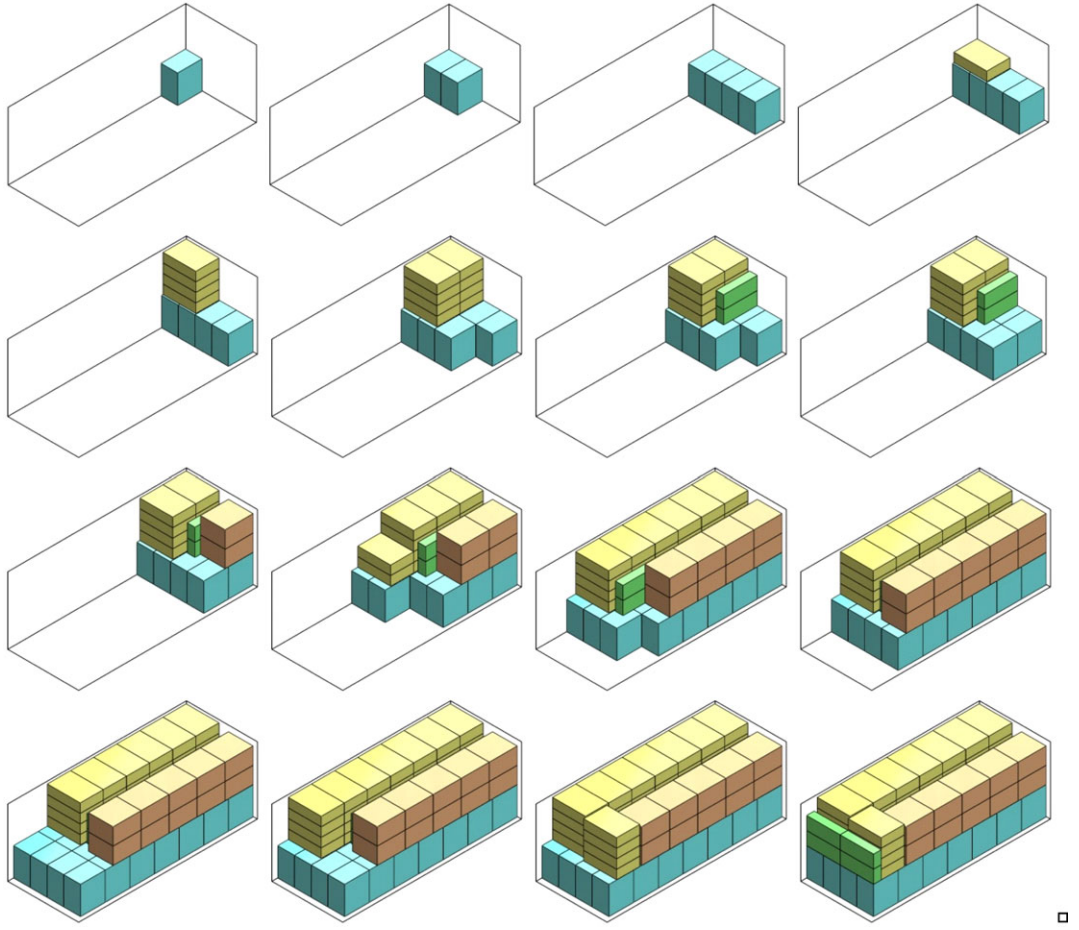


Fig. 10. Example of a cargo loading sequence.

Based on the existing different (un)loading methods it was decided that the loading sequence would follow a wall-building approach, where each box is placed inside the container, starting in the back-bottom-left corner, with boxes placed sequentially along the y -axis and then upward along the z -axis, as illustrated in Fig. 10. This method keeps the box closer to the worker's body prior to lifting or bending, and helps reduce the stress on the shoulders, back, and possibly knees during the reaching and pulling motions, which also contributes to the efficiency of the operation.

4.1. The loading efficiency metrics

To evaluate the efficiency of the loading operation, the average arm's length of the loaded boxes is used as a metric. The *arm's length* of box b can be determined by the difference between x_1 , defined as the maximum x -axis coordinate of a set of boxes N located beneath b , and x_2 , the maximum

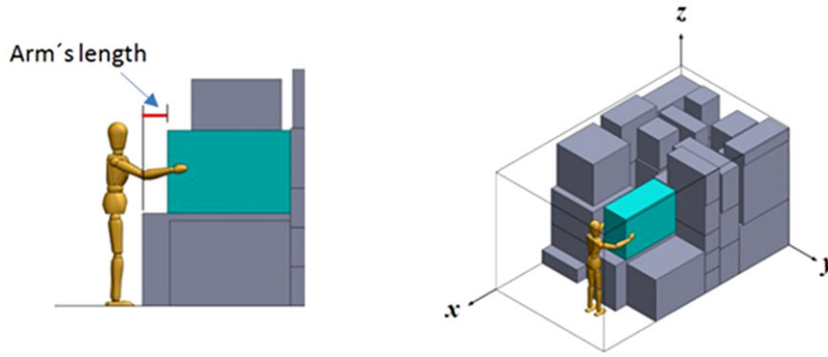


Fig. 11. *Arm's length* metrics.

x -axis coordinate of box b , such that $arm's\ length = \max(x_1 - x_2, 0)$. Let N be determined by all boxes b_j placed beneath b such that

$$(x_b < x_j + d_j) \wedge (x_j < x_b + d_b) \wedge (y_b < y_j + w_j) \wedge (y_j < y_b + w_b) \wedge (z_b \geq z_j + h_j).$$

This reflects the effort made by the operator to load a complete cargo arrangement. Figure 11 presents an illustration of the *arm's length* metric. The *arm's length* value is 0, if there is an overhang in the x -axis direction. There are several other issues related to loading efficiency, as for example the loading weight of the boxes to be lifted, overhead reach, etc. However, since other constraints, such as load bearing, are not considered in the problem, it was decided that the *arm's length* metric would be used, since it is a geometric efficiency metric.

5. The physical packing sequence problem

The physical packing sequence problem (PPSP) uses a loading arrangement M , generated by a container loading algorithm, to determine a sequence by which each box $b_i (i = 1, \dots, n)$ with depth (d_i), width (w_i), and height (h_i) is placed in a predetermined position (x_i, y_i, z_i) inside a container C with depth (D), width (W), and height (H), while at the same time the following additional loading constraints are considered:

- Static stability (Hibbeler, 2010)—Defined as a state where a box is not accelerating in any direction. A box that is currently in equilibrium will remain in equilibrium if two conditions are met:
 - the vector sum of forces applied on the box is zero and;
 - the vector sum of torques that act to rotate the box is zero.
- Loading operation efficiency—The boxes are loaded following a wall-building approach, starting in the back-bottom-left corner of the container while guaranteeing that every box is loaded without vertical and horizontal obstructions, all of its support boxes have been loaded and the arm's length of the box does not surpass a predefined value.

5.1. The PPSA

The PPSP discussed in this paper is solved considering the static stability approach and the loading operation efficiency criteria previously described. The proposed algorithm is presented in Algorithm 4.

The algorithm starts by ordering the set of all the boxes to load M , by nondecreasing order of their x , z and y placement coordinates, respectively. This ordering corresponds to a “wall-building” strategy. The relative position of all boxes can subsequently be found. This allows the support box to be found for each box, and the set of boxes H_i located beneath and behind can be determined for each box i such that

$$\left\{ \begin{array}{l} x_i < x_j + d_j \\ x_j < x_i + d_i \\ y_i < y_j + w_j \\ y_j < y_i + w_i \\ z_i \geq z_j + h_j \end{array} \right. \vee \left\{ \begin{array}{l} y_i < y_j + w_j \\ y_j < y_i + w_i \\ z_i < z_j + h_j \\ z_j < z_i + h_i \\ x_i \geq x_j + d_j \end{array} \right.$$

Algorithm 4. Physical Packing Sequence (Loading arrangement M)

Input: Let M be a set of n boxes b_i with dimensions d_i, w_i, h_i , positioned at coordinates x_i, y_i, z_i , $i = 1, \dots, n$

Output: let k_i be the order position of box b_i , $i = 1, \dots, n$

Begin

Sort M in non decreasing order of coordinates x, z, y

$order_position \leftarrow 1$

Let H_i be the set of boxes beneath and behind box b_i

while ($M \neq \emptyset$) **do**

for $i = 1$ **to** n **do**

if $b_i \in M$ **then**

 ▷ Is the box waiting to be loaded?

if $H_i \cap M = \emptyset$ **then**

 ▷ All boxes beneath and behind have been processed?

 remove b_i from M

$arms_i \leftarrow \text{Call } arms_lenght(b_i)$

 ▷ Determine the *arm's length* of box b_i

if $arms_i \leq arms_limit$ **then**

 ▷ Is the *arm's length* acceptable?

if ($\text{Call Static_Stability}(b_i)$) **then**

 ▷ Is the cargo stable?

$k_i \leftarrow order_position$

$order_position++$

 break

end if

end if

end if

end if

end for

end while

End

Starting with the first box on list M , a box i is selected and it is checked if any of the boxes located beneath and behind are in M , that is, if $H_i \cap M = \emptyset$. If that is not the case, the next box is selected. Then, for the selected box, the *arm's length* is calculated. If the *arm's length* is within the *arm's length* limits, the static stability of the box is checked using the SSA, otherwise the box is removed from the list and a new box is selected. If the box is stable, it is loaded in the container. This procedure is iteratively repeated until there are no boxes in the list. A loading arrangement is considered feasible if all the boxes are loaded in the container.

6. Numerical experiments

In this section, the results of a set of experiments run to evaluate the impact of the PPSA are reported. The algorithm was coded in C and was run on an Intel Core2 Duo at 2.20 GHz computer with 4 GB of RAM.

6.1. Test problem instances

To the best of our knowledge there are no test instances published for the PPSP. For this reason we propose two sets of 15 classes of test instances, with a total of 100 cargo arrangement solutions per class. These are solutions of the well-known instances of Bischoff and Ratcliff (1995) and Davies and Bischoff (1999) provided by the algorithm of Parreño et al. (2008). One of the sets was obtained from Parreño et al. (2008) by enforcing full base support for all boxes, while for the other set the full base support constraint was not considered.

6.2. Computational results

Table 1 shows, for each class of instances, the average results for the 100 arrangements, when the PPSA is applied to container loading solutions. The table presents the average percentage of boxes loaded by the algorithm and the average *arm's length* to which the worker was effectively exposed, in centimeters, for the two sets of solutions previously mentioned. For each set, results were obtained with and without the maximum *arm's length* constraint. A value of 80 cm was considered for this maximum. For the set of instances, where full base support was not enforced, the average percentage of boxes was calculated that would be loaded when a minimum of 85% support for the base of the box was set as a requirement, instead of the stability criterion proposed in this paper.

It was found that if there is no *arm's length* constraint and boxes are fully supported, it is always possible to find a physical packing sequence for the cargo arrangements. It can also be observed that when the *arm's length* constraint is enforced, the impact is similar across instances, with a decrease of around 3% in the feasibility of the loading arrangements.

The results for the test instances, where full base support was not imposed, and was without the *arm's length* constraint, show that an average of around 82% of the boxes were loaded. A decrease in the average percentage of loaded boxes can be observed with the increase in box heterogeneity (which increases from BR1 to BR15). This decrease can also be observed when the *arm's length*

Table 1
Computational experiments

Parreño et al. instances	No arm's length constraint			Arm's length ≤ 80 cm		
	Loaded boxes (%)	Arm's length (cm)	Boxes with at least 85% of base support (%)	Loaded boxes (%)	Arm's length (cm)	Boxes with at least 85% of base support (%)
Full base support						
BR1	100	8.5	-	99	8.0	-
BR2	100	13.5	-	98	12.2	-
BR3	100	11.5	-	98	10.0	-
BR4	100	13.3	-	98	11.8	-
BR5	100	12.9	-	98	11.3	-
BR6	100	13.7	-	97	11.4	-
BR7	100	14.4	-	96	11.8	-
BR8	100	16.2	-	97	13.7	-
BR9	100	15.5	-	96	13.1	-
BR10	100	15.7	-	95	12.5	-
BR11	100	15.1	-	95	11.9	-
BR12	100	14.6	-	95	11.4	-
BR13	100	13.0	-	97	10.7	-
BR14	100	12.3	-	97	10.3	-
BR15	100	12.0	-	97	10.1	-
Mean	100	13.5		97	11.3	
Without full base support						
BR1	97	16.1	96	95	14.5	91
BR2	95	16.2	90	92	13.9	87
BR3	92	19.0	84	88	15.9	81
BR4	92	21.1	83	88	17.8	79
BR5	89	21.2	79	83	16.7	75
BR6	88	21.8	76	82	17.4	72
BR7	85	23.3	73	79	18.3	68
BR8	80	26.5	65	72	20.2	60
BR9	75	26.0	60	68	19.9	55
BR10	74	27.4	57	66	20.4	52
BR11	71	27.1	53	63	20.1	49
BR12	70	27.2	52	62	20.2	48
BR13	70	27.0	50	62	20.3	46
BR14	68	25.3	48	62	19.3	45
BR15	69	24.1	48	62	18.0	45
Mean	82	23.3	68	76	18.2	64

constraint is enforced: in this case the average percentage of loaded boxes decreases from 95% to 62%.

If the stability criterion proposed in this paper was replaced by the partial base support constraint, where at least 85% of the base of the box must be supported, on average only 68% of the boxes would be loaded. This provides evidence that the proposed stability algorithm

is less restrictive than usual static stability measures, while fully guaranteeing cargo static stability.

7. Concluding remarks

In this paper we have proposed two algorithms, a SSA based on the static mechanical equilibrium conditions for rigid bodies, and a PPSA to evaluate the feasibility of cargo arrangements generated by existing CLP algorithms. The new proposed SSA was used within the PPSA, demonstrating its ability to be combined with other algorithms as an evaluation function. Some of the major benefits of the algorithm include its short processing time, the ability to evaluate static stability when the supported area of a box is below 50%, and its proximity to real-world static stability. This capability can lead to the development of new strategies for the generation of cargo arrangements in the CLP.

The computational experiments demonstrate that it is a less restrictive approach to stability than the thumb rules and measures usually found in the literature. For the test instances where full base support is not enforced on the CLP solutions, this algorithm considers that an average of only 82% of the boxes can be loaded safely and stably; when *arm's length* is limited to a maximum of 80 cm only 76% of the boxes can actually be loaded without stability issues.

Acknowledgments

This research was partially supported by the ERDF through the COMPETE Programme, by the Portuguese Government through the FCT (StableCargo—PTDC/SEN-TRA/121715/2010 and PTDC/EGE-GES/117692/2010) and by the Project BEST CASE—SAESCTN-PIIC\&DT/1/2011, which is co-financed by the North Portugal Regional Operational Programme (ON.2-O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF). Special thanks go to Professor F. Parreño who provided his CLP solutions to be used as test instances.

References

- Badler, N., O'Rourke, J., Kaufman, B., 1980. Special problems in human movement simulation. *ACM SIGGRAPH Computer Graphics* 14, 3, 189–197.
- Bhattacharya, A., McGlothlin, J.D., 2012. *Occupational Ergonomics: Theory and Applications* (2nd edn). CRC Press, Boca Raton, FL.
- Bischoff, E.E., Ratcliff, M.S.W., 1995. Issues in the development of approaches to container loading. *Omega* 23, 4, 377–390.
- Bortfeldt, A., Gehring, H., 2001. A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research* 131, 1, 143–161.
- Bortfeldt, A., Gehring, H., Mack, D., 2003. A parallel tabu search algorithm for solving the container loading problem. *Parallel Computing* 29, 5, 641–662.
- Bortfeldt, A., Wäscher, G., 2013. Constraints in container loading: a state-of-the-art review. *European Journal of Operational Research* 229, 1, 1–20.

- Carpenter, H., Dowsland, W.B., 1985. Practical considerations of the pallet-loading problem. *Journal of the Operational Research Society* 36, 6, 489–497.
- Ceschia, S., Schaerf, A., 2011. Local search for a multi-drop multi-container loading problem. *Journal of Heuristics* 19, 2, 275–294.
- Chand, D.R., Kapur, S.S., 1970. An algorithm for convex polytopes. *Journal of the ACM* 17, 1, 78–86.
- Christensen, S.G., Rousøe, D.M., 2009. Container loading with multi-drop constraints. *International Transactions in Operational Research* 16, 6, 727–743.
- Davies, A., Bischoff, E.E., 1999. Weight distribution considerations in container loading. *European Journal of Operational Research* 114, 3, 509–527.
- de Castro Silva, J.L., Soma, N.Y., Maculan, N., 2003. A greedy search for the three-dimensional bin packing problem: the packing static stability case. *International Transactions in Operational Research* 10, 2, 141–153.
- Dibakar, S., Mruthyunjaya, T., 1999. A computational geometry approach for determination of boundary of workspaces of planar manipulators with arbitrary topology. *Mechanism and Machine Theory* 34, 1, 149–169.
- Edelsbrunner, H., Maurer, H.A., 1981. On the intersection of orthogonal objects. *Information Processing Letters* 13, 4/5, 177–181.
- Fuellerer, G., Doerner, K.F., Hartl, R.F., Iori, M., 2010. Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research* 201, 3, 751–759.
- Gehring, H., Bortfeldt, A., 1997. A genetic algorithm for solving the container loading problem. *International Transactions in Operational Research* 4, 5/6, 401–418.
- Gendreau, M., Iori, M., Laporte, G., Martello, S., 2006. A tabu search algorithm for a routing and container loading problem. *Transportation Science* 40, 3, 342–350.
- Gonçalves, J.F., Resende, M.G., 2012. A parallel multi-population biased random-key genetic algorithm for a container loading problem. *Computers & Operations Research* 39, 2, 179–190.
- Haines, E., 1994. Point in polygon strategies. In Heckbert, P. (ed.) *Graphics Gems IV*. Academic Press, San Diego, CA, pp. 24–46.
- Hibbeler, R.C., 1994. *Mechanics of Materials*. Prentice Hall, Englewood Cliffs, NJ.
- Hibbeler, R.C., 2010. *Engineering Mechanics: Statics*. Prentice Hall, Upper Saddle River, NJ.
- Iori, M., Martello, S., 2010. Routing problems with loading constraints. *TOP* 18, 1, 4–27.
- Jarvis, R.A., 1973. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters* 2, 1, 18–21.
- Junqueira, L., Morabito, R., Sato Yamashita, D., 2011. MIP-based approaches for the container loading problem with multi-drop constraints. *Annals of Operations Research* 199, 1, 51–75.
- Junqueira, L., Oliveira, J.F., Carravilla, M., Morabito, R., 2013. An optimization model for the vehicle routing problem with practical three-dimensional loading constraints. *International Transactions in Operational Research* 20, 5, 645–666.
- Kallas, S., 2011. *White paper on transport: roadmap to a single European transport area—towards a competitive and resource-efficient transport system*. Publications Office of the European Union, Luxembourg.
- LaValle, S.M., 2006. *Planning Algorithms*. Cambridge University Press, Cambridge.
- Lin, J.-L., Chang, C.-H., Yang, J.-Y., 2006. A study of optimal system for multiple-constraint multiple-container packing problems. In Ali, M., Dapoigny, R. (eds) *Advances in Applied Artificial Intelligence, Lecture Notes in Computer Science*, Vol. 4031. Springer, Berlin/Heidelberg, pp. 1200–1210.
- Liu, W.-Y., Lin, C.-C., Yu, C.-S., 2011. On the three-dimensional container packing problem under home delivery service. *Asia-Pacific Journal of Operational Research* 28, 5, 601–621.
- Mack, D., Bortfeldt, A., Gehring, H., 2004. A parallel hybrid local search algorithm for the container loading problem. *International Transactions in Operational Research* 11, 5, 511–533.
- Moura, A., Oliveira, J.F., 2008. An integrated approach to the vehicle routing and container loading problems. *OR Spectrum* 31, 4, 775–800.
- Ngoi, B.K.A., Tay, M.L., Chua, E.S., 1994. Applying spatial representation techniques to the container packing problem. *International Journal of Production Research* 32, 1, 111–123.
- Nordbeck, S., Rystedt, B., 1967. Computer cartography point-in-polygon programs. *BIT Numerical Mathematics* 7, 1, 39–64.

- Parreno, F., Alvarez-Valdes, R., Tamarit, J.M., Oliveira, J.F., 2008. A maximal-space algorithm for the container loading problem. *INFORMS Journal on Computing* 20, 3, 412–422.
- Shimrat, M., 1962. Algorithm 112: position of point relative to polygon. *Communications of the ACM* 5, 8, 434.
- Tarantilis, C., Zachariadis, E., Kiranoudis, C., 2009. A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *IEEE Transactions on Intelligent Transportation Systems* 10, 2, 255–271.
- Vukobratović, M., Borovac, B., 2004. Zero-moment point: thirty five years of its life. *International Journal of Humanoid Robotics* 1, 1, 157–173.
- Wäscher, G., Haußner, H., Schumann, H., 2007. An improved typology of cutting and packing problems. *European Journal of Operational Research* 183, 3, 1109–1130.
- Zhu, W., Lim, A., 2012. A new iterative-doubling Greedy Lookahead algorithm for the single container loading problem. *European Journal of Operational Research* 222, 3, 408–417.