# Extending Plant Packing

Matthias Postina, René Schumann, Sonja Preis and Jan Behrens

OFFIS, Escherweg 2, 26121 Oldenburg Germany
matthias.postina|rene.schumann|sonja.preis|jan.behrens@offis.de

**Abstract.** The transportation of plants is expensive, since plants have a comparable high volume in relation to their value. Therefore effective plant packing is an important issue, as the transportation costs are directly dependent on the number of needed trolleys. The potted plant packing problem was introduced in [1] and a first solution was presented. It turned out that the generated plans were not sufficient in comparison to trolleys packed by humans. By observing human packers, we identified plant stacking as a usual habit to increase compactness of the packing. Thus we extended our algorithms, adding the option of stacking plants. In this paper we present the techniques implementing stacking strategies. It is shown that stacking of plants increases the performance of the packing algorithm significantly.

## 1 Introduction

The transportation of plants is expensive, since plants often have a comparable high volume in relation to their value. For standardized transport, potted plants are loaded on transport trolleys (shown in Figure 1). The cost of transportation depends on the number of these trolleys. In order to minimize transportation costs, effective packing of trolleys is necessary. Thus the potted plant packing problem was presented in [2] and steps towards a solution were shown in [1]. Based on the evaluation of test data, it turned out that the computed solutions could not compete with the number of needed trolleys packed by humans. Computed plans need up to 75% more trolleys (see section 4 for details). Different reasons could be identified for this lack of performance.

- Humans commonly use to stack plants on trolley layers.
- The algorithm works on pessimistic data. The size of plants is given in an interval. The algorithm uses always the maximal size, which is a pessimistic assumption.
- Humans pack several layers in parallel.

Based on these results it is obviously necessary to scale the performance of the packing program. In fact stacking of plants is a very efficient way to increase the packing density. For instance the capacity for 10 $l$ pots can be increased by more then 40% using stacking (cf. Figure 5). For other pot sizes the results are similar. So in fact stacking of plants seems to be a promising technique to improve results.

However, the pessimistic assumption about the actual size of the plants can not be relaxed completely. But as plants are a flexible good, this allows to place plants more densely. In consequence the assumptions about the plants dimensions can be relaxed partially. It is assumed that the size of each plant is the mean value of the given interval. When this general rule is not applicable - e.g. during the anthesis of the plants - we assume further that this can be specified for each type of plant. Of course the planning quality as well as the data quality has to be approved and each plant has to be checked to that effect.

The usage of average values for the plants size leads to a more realistic model for packing plants. Moreover the model of plants has to be detailed, to remove some oversimplifications. In our first approach plants and especially their pots were modeled as cylindric objects. It turned out that this model hinders the computation of realistic plans. Real pots are conic, their base has a lower diameter then their top. This allows that pots can be placed more densely at the border area of the trolley layers. This effect is significant for realistic packing. In consequence our model has to be expanded to respect the correct placement of conic pots - hence the base diameter is used for the calculation of the placement at the border area of the trolley layer.

The packing of parallel layers is actually not tackled and remains as future work. But an improvement that avoids to create only marginal used layers does already exist. If such a layer is detected, an algorithm tries to distribute the plants to other layers. This strategy helps to avoid nearly empty layers in general.

Actually two improvements were implemented. First basic data is updated, and the over pessimistic assumptions about the plant size is reduced. Second, stacking of plants is supported.

As the packing problem was described in [2] and a basic solution was presented in [1], this article focuses on the extended feature of stacking and it's integration into the existing tool. Thus the article is structured as follows. In the next section the potted plant packing problem and a solution approach is briefly described. In section 3 the stacking of plants is presented in detail. It is discussed what different techniques exist for stacking and when they can be applied. Then in section 4 case studies are presented, where the computed solutions are compared with human generated packings. Finally we conclude and discuss future work.

## 2 The potted plant packing problem

### 2.1 Problem statement

The major task is to compute a valid packing instruction for a given transportation order. Such a packing instruction must contain directives for the exact placement of each and every plant that is part of the order. Any such directive holds information about the plant's designated place - and each layer's exact placement (mounting height) within the trolley. To clarify the problem, a trolley is shown in Figure 1. A number of further constraints and additional rules have

**Fig. 1.** packed trolleys

to be observed as part of the problem. For example: it is allowed to stack plants on a trolley layer, the placement of layers into trolleys has to respect the stability of trolleys, and the total trolley height usually has to be less than the available internal truck height. Modeling aspects focusing these issues were discussed in [2] and [1].

### 2.2 Existing solution approach

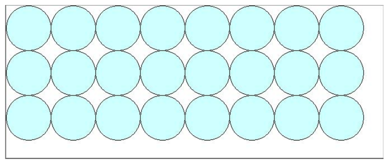
For the sake of simplicity it is assumed that for each and every plant a cylinder can be computed, that contains the plant. To solve the potted plant packing problem it is decomposed into two sub-problems:

- Distribution of plants on trolley layers
- Distribution of layers on trolleys

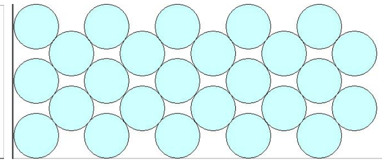
**Packing of layers** The packing of trolley layers corresponds to the packing of circles into a rectangle, since the height of the plants can be omitted. The packing of trolley layers can itself be subdivided into two different tasks.

- packing of equal circles into a rectangle
- packing of unequal circles into a rectangle

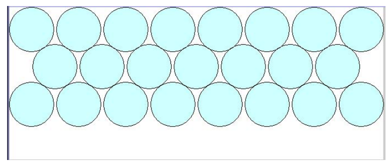
For the packing of equal circles simple heuristics can be applied, that offer already a good solution quality. The circles are placed along the horizontal or vertical of the layer, or are placed in as a grid. Unfortunately it depends on the size of the circles and the rectangle which heuristic performs best. The results of the different placement strategies are shown in Figure 2.



(a) grid placement, 24 circles



(b) placement along the depth 25 circles



(c) placement along the width, 23 circles

**Fig. 2.** regular circle placements

The placement of unequal circles into a rectangle is a quite more difficult task. We adopted the maximum hole degree algorithm (B1.0) presented in [3]. The main idea of this algorithm is the subsequent placement of circles into corners. A corner is defined by two sides of the rectangle, a rectangular side and a circle, or two circles. The first two circles are placed by a simple placement strategy. Then for each circle not placed already within the rectangle, all possible corner placements are computed. The circle being associated with the placement having the minimal distance to another circle or side is chosen next. This is repeated until no more valid corners are found or all elements have been placed. A detailed description of the algorithm as well as a complexity analysis can be found in [3]. An illustration showing the packing of unequal circles into a rectangle is sketched in Figure 3.

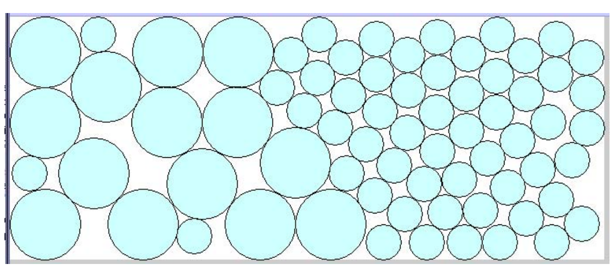


**Fig. 3.** placement computed by the maximum hole degree algorithm

Even if the trolley layers are computed step by step the design decisions within the algorithm are taken into a direction that the overall summed height of all layers is minimized. This is an heuristic approach aiming at the minimization of the number of needed trolleys. Consequently the objective function regards the global context even though this computation is separated into partial blocks.

**Packing of trolleys** The problem of distributing the layers to trolleys is a classical bin packing problem. But additionally the position of each layer within a trolley has to be computed. Trolley layers are hooked into mounting points, which are generally found at 5 cm intervals from a base of 20 cm up to a height of 190 cm. The placement of layers within a trolley follows a simple strategy:
The tallest trolley layer is hooked into the topmost mounting point that still ensures the adherence of all other constraints, especially the maximum allowed height. All remaining trolley layers are then sorted in ascending order by their weight and inserted top to bottom into the trolley. This strategy aims at two goals. It tries to

– maximize the usage of available space on the truck and
– lower the center of gravity to the nethermost point for stability reasons.

## 3 Stacking plants

As already mentioned solutions for the potted plant packing problem generated by humans are outperforming computed solutions. Contrary to programs, human packers have an intuitive understanding of spatial optimization, so stacking plants was observed as a typical human packing habit. To imitate such stacking is consequently the first approach to improve the packing algorithm. This section categorizes the plants stacking problem as the stacking on homogeneous patterns and as the stacking on heterogeneous patterns.

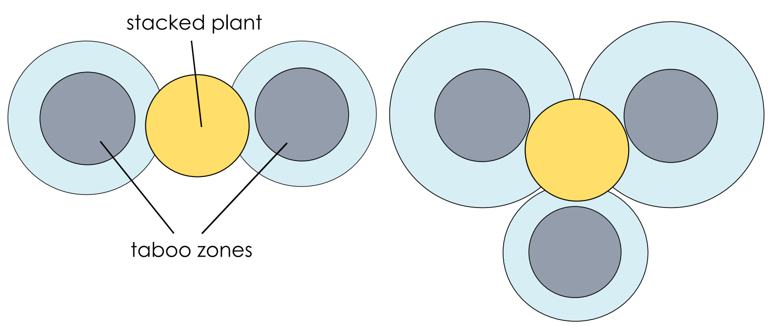
### 3.1 Stacking on homogeneous patterns

The terms homogeneous and heterogeneous pattern are related to a layer of a trolley which is already initially packed. Whenever a layer of a trolley is homogeneously packed - which means only plants of the same category with exactly the same pot and plant size were used to fill the basic layer - we say this layer follows a homogeneous pattern. Figure 2 has already shown such homogeneous patterns. In order to support stacking, we need to upgrade the planning algorithms and we have to regard stacking in more detail.

**Definition 1:**
We call a position a *stackable place* whenever a pot could be placed on the top of a minimum set of three neighbored pots having the same height.

We chose this minimal set of three plants as bed for a stacked plant since plants are fragile goods and falling over leads to a total loss of plant value. So the configuration pictured in Figure 4 (a) would be an illegal configuration according to our definition. For the same reason stacking needs to take care of the underlying plants. Keeping this in mind, we defined a taboo zone for each plant reflecting its actual shape. These taboo zones have to be respected when looking for valid stackable places. Since we focus on the packing of circular objects it is sufficient to store the center and the radius to indicate a stackable place. Furthermore we need a statically defined minimal footprint to guarantee a sufficient contact surface and also a maximal boundary to prevent damage from the underlying plants. On the strength of stability we also restrict the stack height to a maximum of two per trolley layer. Basically this reflects the packing habit of human packers.

**Definition 2:**
We say a stackable place is *valid* for a certain pot category, if the footprint of pots assigned to this category guarantees a sufficient contact surface without violating the taboo zones of the underlying plants.




(a) Non stackable place.

(b) Valid stackable place for the stacked plant (assuming same pot hight).

**Fig. 4.** Visualization of Definitions

It is easy to recognize the set of stackable places when the basic layer follows a homogeneous pattern. One needs to identify the size of the radius, the size of the taboo zone and the used placement strategy (cf. Fig. 2). By means of this information one can calculate an offset along a direction vector pointing out of each underlying circle center. So finding the center of the stackable places could be seen as a pattern/raster shift of the underlying layer. Such a pattern shift is shown in Figure 5 where the center of each stackable place is moved along the direction vector $\overrightarrow{v}$.

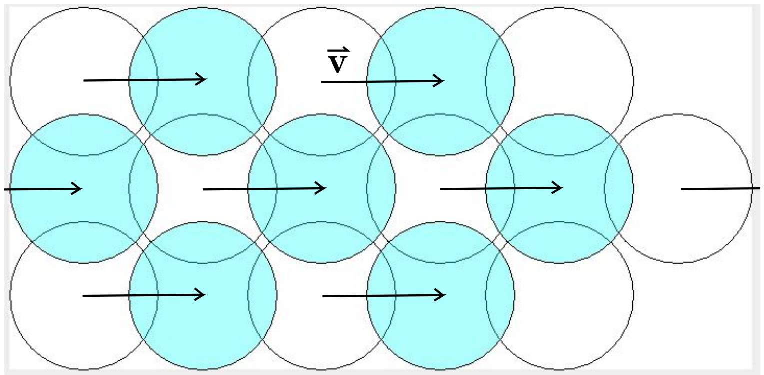



**Fig. 5.** Pattern postponement along a direction vector v (10 $l$ pots following a homogeneous pattern)

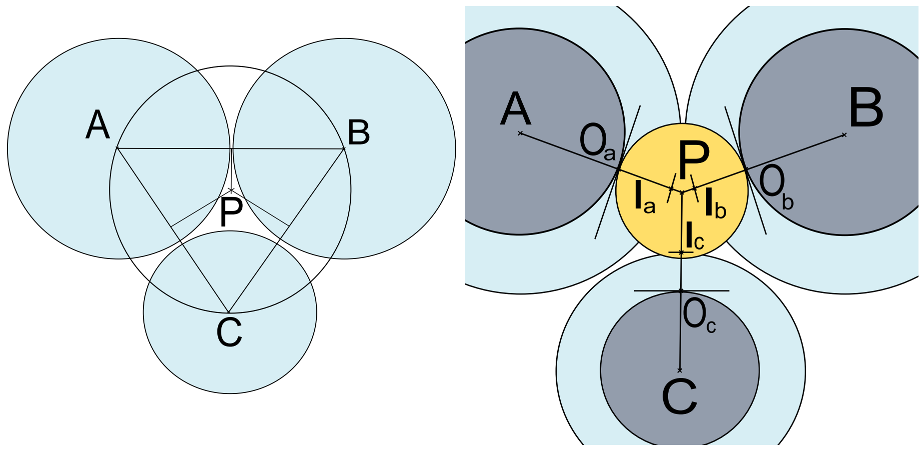
### 3.2 Stacking on heterogeneous patterns

In analogy to homogeneous pattern, we say a layer follows a heterogeneous pattern, when plants of different categories (with miscellaneous pot diameters) and different pot heights were used to fill the basic layer of a trolley. Figure 3 shows such a heterogeneous arrangement.

Obviously it is much harder to spot stackable places on the top of a layer following a heterogeneous pattern. Since pot heights may vary one need to identify regions in the layer where at least three neighbored pots are having the same pot height. Furthermore these three pots need to be grouped in a way that offers a valid stackable place - which is close enough to allow stacking but wide enough to respect taboo zones of the underlying plants. Moreover one needs to calculate the centers of the stackable places for each possible triangular constellation individually. Such a center of a stackable place could be calculated as follows:

When regarding a troika of pots of the same height, the centers $A, B$ and $C$ of these three circular objects are forming a triangle. We want to find a point $P$ having the same distance to each of $A, B$ and $C$. Such $P$ is the center of the circumscribed circle of the triangle which is the intersection point of the perpendicular bisectors of the sides. This is shown in Figure 6 (a). Now, one need to show that the identified stackable place having $P$ as center is a valid stackable place for a certain pot category according to Definition 2. In order to do so, we will formalize Definition 2 based on the sketch in Figure 6 (b).

Given are the points $A, B$ and $C$ as centers of the three neighbored pots $C_a, C_b, C_c$ with known radii. All of these pots are planted and each taboo zone is known and identified by a taboo radius. We already calculated point $P$ and define $\overline{PA}, \overline{PB}$ and $\overline{PC}$ as straight lines - each going through $P$ and one of the centers $A, B$ and $C$. Further we name the intersection points of these lines with the taboo




(a) Finding P as center of a possible stackable place

(b) Identifying the category of the stackable place

**Fig. 6.** Finding stackable places

circles $O_i$ and the intersection points of these lines with the pot bounding circles $I_i$ $(i \in a, b, c)$.

To reduce search space we introduced the concept of categories in [1]. Still following this concept, each plant is classified as member of a specific cylindric category. Such a category could be seen as a virtual box hosting plants with similar dimensions. The radius $R_c$ of the cylinder is taken to define a valid stackable place for a certain pot category.

We say the stackable place having $P$ as center is a valid stackable place for a certain pot category $Cat_n$ $(n \in \mathbb{N})$ , if

$R_c \leq \min[\left|\overline{PO_a}\right|, \left|\overline{PO_b}\right|, \left|\overline{PO_c}\right|]$ *and*

$R_c \geq \max[\left|\overline{PI_a}\right|, \left|\overline{PI_b}\right|, \left|\overline{PI_c}\right|]$.

We define the set of stackable places $S$. Valid stackable places are assigned to pot categories so that each category holds a subset $V_n$ of $S$, $V_n \subseteq S$ ($n \in \mathbb{N}$ and $n$ is corresponding to the index of the categories $Cat_n$). The data structure $SP$ is a hash holding the key value pairs of categories and subsets $V_n$ of $S$ :

$$
\begin{aligned}
SP = \{ \quad & Cat_1 => V_1 \quad , \\
& Cat_2 => V_2 \quad , \\
& ... \quad , \\
& Cat_n => V_n \quad \}
\end{aligned}
$$

Further we have a second hash $CS$ holding the key value pairs of categories $C_n$ and corresponding plant sets $Pl_n(n \in \mathbb{N})$ which are subsets of all non placed plants:

$$
\begin{aligned}
CS = \{ \quad & Cat_1 => Pl_1 \quad , \\
& Cat_2 => Pl_2 \quad , \\
& ... \quad , \\
& Cat_n => Pl_n \quad \}
\end{aligned}
$$

The stacking algorithm formalized in Algorithm 1 works as follows: Given the hashes $SP$ and $CS$, the algorithm starts to assign plants in ascending order with respect to their plant category (which means highest dimension first, ordered by the height) to stackable places ($assign(...)$). During this processing, the hash $SP$ - representing valid stackable places per category - is updated permanently ($updateV_x(...)$). This leads to a shrinking set of valid stackable places. Whenever a plant is assigned to a stackable place, such plant is deleted from $CS$ ($removeElementFromPl_j(...)$). The algorithm stops the stacking for a trolley layer, when the hash $SP$ is empty or when all plants were placed. Figure 7 shows a graphical representation of an exemplary calculation.

1. $SP$ `// hash, given as described`
2. $CS$ `// hash, given as described`
3. $V_i$ `// subset of` $S$
4. $Pl_i$ `// subset of all non placed plants`
5. `WHILE (` $\exists i \in \mathbb{N} : V_i \neq \emptyset \vee Pl_i \neq \emptyset$ `)`
   - $j = \min\{i \in \mathbb{N} : V_i \neq \emptyset \wedge Pl_i \neq \emptyset\}$
   - $assign(v_{jx}, pl_{jy})$ `// (` $x, y \in \mathbb{N}, v_{jx} \in V_j, pl_{jy} \in PL_j$ `)`
   - $removeElementFromPl_j(pl_{jx})$
   - $updateV_x(v_{jy})$

Algorithm 1: Stacking on heterogeneous patterns

## 4 On the influence on stacking

After we have implemented the aforementioned stacking strategies, we compare the computed solutions with and without stacking to solutions generated by humans. As we improved only one aspect of the previous mentioned it is not surprisingly, that humans still need fewer trolleys. But as can be seen in table 1 the quality of the solutions with stacking increases.
Scenario 1 comprises homogeneously and heterogeneously packed layers. Scenario 2 is more difficult, as there exist no homogeneous layers. Due to the aforementioned model extension concerning the form of pots, the homogeneously packed

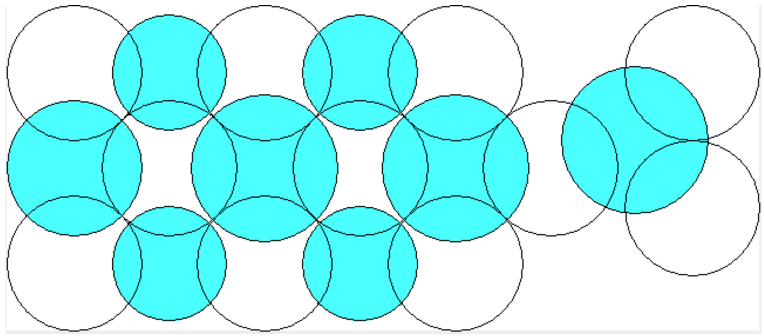


**Fig. 7.** Stacking on heterogeneous patterns

layers are equal either packed by hand or computed. So the quality difference can be explained by different packed heterogeneous layers and differences between the assumed and real height of the layers. For this reason the quality of the schedules in scenario 1 is better then in scenario 2.
Nevertheless in both cases stacking can improve the computed results drastically. Of course the usage of percent values is problematic, as the number of CC-trolleys is a discrete value. But the improvement by stacking is as significant in absolute as in relative numbers. Analyzing the trolleys packed by humans we

| scenario | #CC needed | without stacking | | | with stacking | | |
|---|---|---|---|---|---|---|---|
| | | #CCs | abs. delta | delta in % | #CCs | abs. delta | delta in % |
| 1 | 5 | 7 | 2 | 40% | 6 | 1 | 20% |
| 2 | 8 | 14 | 6 | 75% | 10 | 2 | 25% |

**Table 1.** Results of the Case Study of effective packing

had to state that human packers take advantage of the fact that they are breaking rules the planning strategy has to respect. For example plants were stacked up to three levels per trolley layer or plants were stacked on the base of only two plants. So it was possible to place more plants on one layer and to pack the plants more densely. Of course such constraint violation is not thinkable for the algorithm. It is actually in discussion that the packing rules should be applied for humans as well to avoid plant damages.

## 5 Summary and Future work

When observing human packers it turned out that plant stacking reflects a usual habit to increase compactness of the packing. By know our planning algorithm was not implementing such a case. This paper shows our approach to enable the algorithm to support plant stacking. We distinguished between stacking on homogeneous and heterogeneous patterns and indicated solutions for both of them. It is assumed that stacking will have a relevant impact on the quality of packing solutions. First test results are reinforcing such assumption.

For the sake of simplicity we are dealing with plant categories rather than individual plants. However, the categories are either boxes or cylinders whereas a plant pot is usually tapered. So contrary to cylinders the top and bottom diameters of the pots are unequal. Further research will have to detect if such a simplification leads to proper results or if the data model needs to be updated. Early tests using such data models and limited to homogeneous patterns are indicating, that computed solutions were on a par with solutions generated by human packers in such cases.

Inspired by human packing habits for a second time, the parallel packing of trolley layers is another field of further research. As described, the objective function (minimization of the overall height of all trolleys) is already designed globally but the algorithm works in layers. Using parallel packing one may assume better results, but with respect to complexity and the corresponding runtime of the algorithms we would like to watch more real life results of this approach first. This is because the program based on the planning algorithms is also used in a real time context: In order to be able to calculate the associated shipping costs the dispatcher needs to know how many trolleys are necessary to fulfill an order, which is given by phone at that moment.

## References


1. Schumann, R., Behrens, J.: The potted plant packing problem: Towards a practical solution. In Sauer, J., Edelkamp, S., eds.: 20. Workshop Planen, Scheduling und Konfigurieren, Entwerfen (PUK), Bremen, Universität Bremen (2006) 37 – 48
2. Schumann, R., Behrens, J., Siemer, J.: The potted plant packing problem. In Sauer, J., ed.: 19. Workshop Plan, Scheduling und Konfigurieren / Entwerfen (PUK). Fachberichte Informatik, Koblenz, Universität Koblenz-Landau Fachbereich Informatik (2005)
3. Huang, W.Q., Li, Y., Akeb, H., Li, C.M.: Greedy algorithms for packing unequal circles into a rectangular container. Journal of the Operational Research Society (JORS) **56** (2005) 539 – 548