1 Description of the algorithm

1.1 Assumptions

1. Goal of the algorithm: minimizing the number of pallet load units.
2. The basis for planning is the PRODUCT_ID set in the order, the number of which is smaller than the standard number.
3. The number of PRODUCT_ID units in a shipment (WMS) is not subject to division.
4. Adding subsequent PRODUCT_ID layers to the mix takes into account:
    - height (H) of the layer,
    - degree of layer filling (fw),
    - fragility (stackability) class parameter (CL),
    - weight (W).

1.2 Input data

- order (ORDER_ID)
- place of shipment (DB)
- product name (PRODUCT_ID)
- of product units in the order (BU_QUANTITY)
- number of product units per standard volume (Quantity_Max)
- number of layers in the normative pjl (Layers_Norm)
- product's fragility class (CAPACITY_LOAD_CLASS)

1.3 Output

- pallet loading unit number (LOADUNIT_ID)
- weight of pallet loading unit (WEIGHT) with carrier [mm]
- height of pallet loading unit (HEIGHT) with carrier [kg]
- type of pallet loading unit (TOP or BASE) due to the possibility of creating sandwich pallets (SNDW)

1.4 Brief description of the algorithm

The algorithm works for a set of orders that come from at least one contractor. Products are assigned to pallet loading units separately for each order.

The following steps are performed for subsequent orders from the set:

1. For each assortment (Product_ID), the number of full layers and the number of residual layers are determined. For the non–full layers, it determines the value of the layer filling factor fw.
2. Full layers are placed on pallets while maintaining the permissible pallet weight and pallet height. First, homogeneous pallets. Next, mixed pallets. In both cases taking into account the fragility class of the goods. Pallets that have reached the permissible values of weight or height parameters (W^{max}, H^{max} respectively) are BASE pallets. The remaining pallets will be used to arrange the non–full layers. If all goods from a given order have been assigned to pallet loading units, go to the next order.
3. Non–full layers are then considered:
   - They are tried to be selected so that the difference in the height of the two or three assortments that are potentially to constitute it does not exceed 10 [mm] and the total fill factor of the layer created from them is less than 1.0. Then they are added to the remaining pallet created in step 1, taking into account the weight and maximum height values. If there are no longer pallet loading units with full layers, we create new mixed pallet loading units, taking into account the fragility class of the goods and the weight and height of the pallet. The pallet loading units created in this way are of the BASE type.
   - If there are no goods whose height difference does not exceed 10 [mm], they are combined to form a layer whose total fill factor is not greater than 1.0 and added as the highest layer of the pallet loading unit, taking into account the weight and maximum height of the pallet . The created pallet loading unit is of the TOP type.
4. We combine the remaining products (Product_ID) into layers whose total fill factor does not exceed 1.0. We create a new pallet loading unit and add a layer to it, taking into account the weight and maximum height values. The created pallet loading unit is of the TOP type.
5. When all Product_IDs from the current order have been assigned to pallet load units, the degree of weight and height balance of the created mix type pallet load units is checked. If satisfied, the algorithm terminates for the current order. If they are not, the values of the W^{max} and H^{max} parameters are decremented and the algorithm returns to step 2. In the pseudocode below, the parameter values are decremented to 0.99 of their current value.

1.5 Indexes

- i – product included in the order, where $i = 1, 2, \ldots n, \ldots I$
- j – pallet mix type loading unit
- w – layer on a pallet mix loading unit
- z – client's order

1.6 Parameters

- CL_i – fragility class of the ith product (according to the standard)
- CL_j – fragility class of the jth pallet loading unit determined as the highest CLi value of the PRODUCT_ID items included in the jth pallet loading unit mix
- fw – layer filling degree; parameter subject to control $fw =< 0.7; 1 >$
- fW_{top} - the degree of filling of the highest pallet layer of a load unit
- H_i – unit height of the ith product (according to the standard)
- H_j – height of the jth pallet load unit mix
- ^{Hmax} – maximum height of a pallet load unit mix (system parameter)
- H^{diff} – difference between the maximum height of the pallet loading unit mix $Hmax$ and the height of the jth pallet loading unit mix H_j
- LP_i – number of units of the ith product in the order
- LP_i^{nor} – number of units of the ith product on a standard pallet loading unit
- LW_i^{nor} – number of layers of the ith product in a standard pallet loading unit
- S_j – status of the $j-th$ pallet loading unit mix due to the possibility of being included in the planning of the construction of the SNDW T – pallet loading unit TOP, i.e. it can only be placed on the top of the SNDW; B – BASE pallet loading unit, i.e. can be placed as a base or intermediate layer
- w_i – number of layers of the ith product in the order, where $w_i = w_i^{INT} + w_i^R$
- w_i^{INT} – total value of the number of layers of the ith product in the order
- w_i^R – non-integer value of the number of layers of the ith product in the order
- W_i – weight of the ith product in the order
- W_i^{norm} – standard weight of a pallet loading unit with the ith product
- W_j – weight of the jth pallet load unit mix
- W_j^{zpk} – the difference in the weight of the standard pallet loading unit with the ith Winorm product and the weight of the jth mix pallet loading unit, determined when breaking the fragility class parameter
- W - a set of single layers opened with 2 or 3 Product_ID. Each layer can be supplemented with another Product_ID and added as the highest layer in the palette

- W_n – weight of n products in the order
- W^{max} – maximum weight of a pallet load unit mix (system parameter)
- W^{diff} – difference between the maximum weight of the pallet loading unit mix W^{max} and the weight of the jth pallet loading unit mix W_j
- ZP – a set of products for planning a pallet load unit mix

1.7 Suggested parameter values

- $H^{max} = 1700[mm]$
- $W^{max} = 700[kg]$
- $f_w \in [0.7\ 1.0]$

1.8 Operators

- $\cup$ - Adding the product layer or Product_ID to the top layer of the palette that is currently being built. Updates the palette parameters. The goods are marked as assigned to the pallet - they become planned. The algorithm stores information about the Product_ID state in the *Added_i* array.
- $\sqcup$ - Binary operator. Returns true if the $Product\_ID_i$ and $Product\_ID_{i+n}$ items were combined in one layer. False otherwise.
- $\uplus$ - Adding a layer or Product_ID to the palette that is currently being built taking into account the fragility class. Updates the palette parameters. The goods are marked as assigned to the pallet - they become planned.
- $\setminus$ - Removing the last added layer or Product_ID from the palette that is currently being built. Updates the palette parameters. The goods return to the pool of goods from the current order to be planned - they become unplanned. The algorithm stores information about the Product_ID state in the *Added_i* array.
- $\cap$ - Determining the value os layer filling factor for at least two Product_IDs included in the layer.

2 Algorithm

Algorithm 1 Palet Stacking algorithm

Input: $z, CL(i), ZP, H^{max}, W^{max}, f_{wmin}, f_{wmax}, h_{diff}$
Output: $SetSofpalletlabeledas, , TOP''or, , BASE''$

1: $j \leftarrow 0$
2: $S \leftarrow \emptyset$
3: **for each** $z \in$Z **do** ▷ Execute for each order in the set of orders
4: **for each** $Product\_ID \in z$ **do**
5: $l_i^{INT} \leftarrow int(\frac{LP_i}{LP_i^{NOR} * LW_i^{NOR}})$
6: $l_i^R \leftarrow \frac{LP_i}{LP_i^{NOR} * LW_i^{NOR}} - l_i^{INT}$
7: $l_i^{Planned}(1 : l_i^{INT} + l_i^R) \leftarrow false$
8: $Added_i \leftarrow false$ ▷ Each Product_ID in the order has not been assigned to a pallet loading unit
9: $l_i = l_i^{INT} + l_i^R$
10: $fw_i^{INT} \leftarrow 0$ ▷ The fill factor of the full layer for a given Product_ID
11: $fw_i^R \leftarrow 1 - l_i^R$ ▷ The fill factor of the non–full layer for a given Product_ID
12: **end for each**
13: **for each** $i \in$z **do**
14: **if** $fw_i^{INT} = 0$ **and** $fw_i^R = 0$ **then**
15: $S \leftarrow S \cup Make\_full\_layers\_pallet(i, CL(i), H^{max}, W^{max}, j)$ ▷ Creating pallet loading units from full layers
16: $Added_i = true$
17: **go to** 22
18: **else**
19: **go to** 22
20: **end if**
21: **end for each**
22: **for each** $i \in$z **do**
23: $sort(i, CL(i), 1)$ ▷ Product_IDs that have not been assigned to a pallet loading unit are sorted ascending by fragility class
24: $sort(i, CL(i), w_i, -1)$ ▷ Within each fragility class we sort in descending order by the number of layers that have not been assigned to a pallet unit load
25: **end for each**
26: **for each** $i \in$z **do**
27: **if** $\exists\, i : l_i^{Planned}(:) = false \& l_i^{Planned}(:) = true$ **then** ▷ Are there any items on the pallet load unit to which Product_ID has not been attempted?
28: **go to** 50
29: **else**
30: **if** $\exists_i : l_i^{Planned}(1 :) = false$ **then** ▷ Are there Product_IDs in the order that have not been attempted to be added to the Product_IDs that make up the mix type pallet unit load?

go to 74
else
for each $j \in$S **do**
if $|W_j - W_{\neq j}| < W^{diff} \& |H_j - H_{\neq j}| < H^{diff}$ **then** ▷ Check the degree of weight and height balance of pallet load units
go to 751
else
$W^{max} \leftarrow 0.99 * W^{max}$ ▷ Changing the values of the W^{max} and H^{max} parameters to achieve improved balance
$H^{max} \leftarrow 0.99 * H^{max}$
go to 13
end if
end for each
end if
end if
for each $j \in$S **do** ▷ Is it possible to add new $Product\_ID$ to already created pallet loading units?
$W_j^{mix} \leftarrow \frac{\sum_{j=1}^{J} W_j}{W_{max}}$ ▷ Calculate the weight coefficient W_{mix} and the height H_{mix} of a pallet mix loading unit.
$H_j^{mix} \leftarrow \frac{\sum_{j=1}^{J} H_j}{H_{max}}$
end for each
$W^{max} \leftarrow max_j(W_j^{mix})$ ▷ Choose a pallet with the maximum weight or height
$H^{max} \leftarrow max_j(H_j^{mix})$
for each $j \in$S **do**
$W_j^{diff} \leftarrow W^{max} - W_j$ ▷ Calculate the difference in the maximum weight and height of the pallet loading unit j and the maximum weight and height, respectively
$H_j^{diff} \leftarrow H^{max} - H_j$
end for each
for each $i \in$z **do** ▷ Select $Product\_ID$ with a weight or height not greater than the differences calculated above
if $W_i <= W^{diff} \| H_i <= H^{diff}$ **then**
go to 62
else
$S_j \leftarrow, , BASE''$ ▷ The BASE type pallet loading unit was built
go to 26
end if
end for each
for each $j \in$S **do** ▷ For a pallet loading unit, calculate the difference between its weight W_j or height H_j and the maximum weight W^{max} and height H^{max}, respectively
$H_j^{diff} = H^{max} - H_j$
$W_j^{diff} = W^{max} - W_j$

65: **for each** $i \in$z **do** ▷ Are there Product_IDs in the order whose height or weight is less than the difference calculated above (H_{diff}, W_{diff} respectively)
66: **if** $H_i <= H_{diff} \| W_i <= W_{diff}$ **then**
67: **go to** 236
68: **else**
69: $S_j \leftarrow, , BASE''$ ▷ The BASE pallet load unit is builded
70: **go to** 26
71: **end if**
72: **end for each**
73: **end for each**
74: **for each** $i \in$z **do** ▷ Are there any Product_ID items that have not been combined yet and have full layers, i.e. $fw \in [0.7; 1.0]$?
75: **if** $Added_i = false \& fw_i \in [0.7; 1.0]$ **then**
76: **go to** 82
77: **else**
78: **go to** 175
79: **end if**
80: **end for each**
81: $count \leftarrow 0$ ▷ Are there at least two Product_ID items in the order that have full tiers ($fw \in [0.7; 1.0]$)
82: **for each** $i \in$z **do**
83: **if** $fw_i \in [0.7 : 1.0]$ **then**
84: $count \leftarrow count + 1$
85: **end if**
86: **end for each**
87: **if** $count >= 2$ **then**
88: **go to** 92
89: **else**
90: **go to** 143
91: **end if**
92: **for each** $i \in z \& fw_i \in [0.7 : 1.0]$ **do** ▷ Is it possible to combine two Product_IDs into a layer? The difference in their heights $\|H_i - H_{i+n}\| <= 10[mm]$
93: **if** $\exists_n : \|H_i - H_{i+n}\| <= 10$ **then** ▷ Will adding the combined Product_ID to a pallet unit load not exceed the value of W^{max} or H^{max}?
94: **if** $\sum_{k=1}^{I} W_k > W^{max} \| \sum_{k=1}^{I} H_k > H^{max}$ **then**
95: **go to** 74 ▷ The values of W^{max} or H^{max} have been exceeded
96: **end if**
97: **if** $\sum_{K=1}^{I} W_k \leq W^{max} \& \sum_{k=1}^{I} H_k <= H^{max}$ **then**
98: **if** $fw_i + fw_{i+n} > 1.0$ **then** ▷ Checking the fill factor of the common layer
99: **go to** 74
100: **else**

```
101:                    if ∃_i : planned_i = false then                ▷ Is there a Product_ID in the order that is not assigned to a pallet loading unit?
102:                        for each j ∈ S do
103:                            if W_j + W_i ≥ W_max‖W_j + W_i ≥ W_max‖CL_i ≥ CL_j then   ▷ Possibility of adding to an existing mix pallet loading unit due to values of W^max, H^max and fragility class
104:                                if fw_j + fw_i ≤ 1 then
105:                                    S_j ∪ Product_ID(S_j, i)
106:                                    if fw_i + fw_{i+n} < 0.7 then      ▷ If the total fw of a layer ¡ 0.7, split the pre-connected Product_IDs that constitute this layer
107:                                        i \ i + n
108:                                        go to 74
109:                                    else
110:                                        S_j = BASE                     ▷ A pallet loading unit has been created. It is of the BASE type
111:                                        go to 30
112:                                    end if
113:                                else
114:                                    if fw_i + fw_{i+n} < 0.7 then      ▷ If the total fw of a layer ¡ 0.7, split the pre-connected Product_IDs that constitute this layer
115:                                        i \ i + n
116:                                        go to 74
117:                                    end if
118:                                    if fw_i + fw_{i+n} ∈ [0.7; 1.0] then   ▷ A pallet loading unit has been created. It is of the BASE type
119:                                        S_j = BASE
120:                                        go to 30
121:                                    end if
122:                                end if
123:                            else
124:                                if fw_i + fw_{i+n} < 0.7 then  ▷ If the total fw of a layer ¡ 0.7, split the pre-connected Product_IDs that constitute this layer
125:                                    i \ i + n
126:                                    go to 74
127:                                else
128:                                    if fw_i + fw_{i+n} ∈ [0.7; 1.0] then   ▷ A pallet loading unit has been created. It is the BASE type
129:                                        S_j = BASE
130:                                        go to 30
131:                                    end if
132:                                end if
133:                            end if
134:                        end for each
```

```
                    else
                    end if
                end if
            end if
        else
            go to 30                                   ▷ If the height difference ‖H_i − H_{i+n}‖exceeds10[mm].
        end if
    end for each
    if i : fw_i ∈ [0.7; 1.0]&∃_n CL_i < CL_{i+n} then   ▷ Can further items from the order be placed on the full-layer i Product_ID item due to the fragility class (CL_i ≤ CL_{i+n})
        if fw_i + fw_{i+n} <= 1 then
            if Σ_{i=1}^{I} W_i < W^{max} & Σ_{i=1}^{I} H_i < H^{max} then ▷ If the weight and height of the pallet loading unit do not exceed the maximum values after entering Product_IDs, add Product_IDs
                if fw < 0.7 then                       ▷ Check the layer feeling degree of top layer
                    S_j ∪ Product_ID(S_j, i)
                    S_j = TOP
                    go to 30
                end if
                if fw ∈ [0.7; 1.0] then
                    S_j ∪ Product_ID(S_j, i)
                    if Try_Pillars(S_j) = true then
                        S_j ← TOP
                        go to 30
                    else
                        S_j ← BASE
                        go to 30
                    end if
                end if
            end if
        end if
    else                                               ▷ If you cannot Product_ID with the full layer fw ∈ [0.7; 1.0] creates a new mix type pallet loading unit
        Make_new_Pallet(Product_ID_i)
        j ← j + 1
        if fw_i < 0.7 then                             ▷ Check the layer feeling degree of top layer
            S_j = TOP
            go to 30
        end if
```

170: **if** $fw_i \in [0.7; 1.0]$ **then**
171: $S_j = base$
172: **go to** 30
173: **end if**
174: **end if**
175: **for each** $i \in z$ **do** ▷ Check if there are Product_IDs in the order that can create a layer
176: **if** $\exists_n : \|H_i - H_{i+n}\| <= 10$ **then** ▷ Condition for combining two Product_IDs items into a layer in order to be added to a pallet load unit
177: **go to** 182
178: **else**
179: **go to** 30
180: **end if**
181: **end for each**
182: $W^{sum} \leftarrow 0$ ▷ Check whether the weight of the pallet loading unit after adding the Product_IDs items that can be combined does not exceed the maximum W^{max}
183: $top\_layer \leftarrow \emptyset$
184: $fw_{top\_layer} \leftarrow 0$
185: **for each** $i \in z$ **do**
186: **if** $\|H_i - H_{i+n}$ ¡ 10 **then** ▷ If combining is possible
187: $top\_layer \leftarrow top\_layer \cup i + n$
188: $W^{sum} \leftarrow W^{sum} + W_{i+n}$
189: $fw_{top\_layer} \leftarrow fw_{top\_layer} + fw_{i+w}$
190: **end if**
191: **end for each**
192: **if** $W^{sum} + W_j < W^{max}$ **then** ▷ If it does not exceed, add combined Product_IDs
193: **if** $fw_{top\_layer} < 1 \& H_j + H_{i+n} < H^{max}$ **then**
194: $Create\_Pillars()$ ▷ Consider building pillars
195: **end if**
196: **if** $fw_{top\_layer} < 0.7$ **then**
197: $S_j \cup top\_layer$
198: $S_j = TOP$
199: **go to** 175
200: **end if**
201: **if** $fw \in [0.7; 1.0]$ **then**
202: $S_j \cup top\_layer$
203: **if** $try\_pillars(S_j) = true$ **then** ▷ Are there any pillars in the top layer?
204: $S_j = TOP$
205: **go to** 175

206: **else**
207: $S_j = BASE$
208: **go to** 175
209: **end if**
210: **end if**
211: **else**
212: **while** $W^{sum} + W_j > W^{max}$ **do** ▷ If it exceeds, we remove items from the added Product_IDs until the weight does not exceed W^{max}
213: $top\_layer \leftarrow top\_layer \setminus i + n$
214: $W^{sum} \leftarrow W^{sum} - W_{i+n}$
215: $fw_{top\_layer} \leftarrow fw_{top\_layer} - fw_{i+w}$
216: **end while**
217: **if** $fw_{top\_layer} < 1 \& H_j + H_{i+n} < H^{max}$ **then** ▷ Check whether the filling factor of the top layer is not greater than 1 and whether the maximum height H^{max} has not been exceeded
218: $Create\_Pillars()$ ▷ Consider building pillars
219: **end if**
220: **if** $fw_{top\_layer} < 0.7$ **then** ▷ Check the filling factor of the top layer
221: $S_j \cup top\_layer$
222: $S_j \leftarrow TOP$
223: **go to** 175
224: **end if**
225: **if** $fw \in [0.7; 1.0]$ **then**
226: $S_j \cup top\_layer$
227: **if** $try\_pillars(S_j) = true$ **then** ▷ Are there any pillars in the top layer?
228: $S_j \leftarrow TOP$
229: **go to** 175
230: **else**
231: $S_j \leftarrow BASE$
232: **go to** 175
233: **end if**
234: **end if**
235: **end if**
236: $CL_j \leftarrow 0$ ▷ Check the fragility class of the pre-built pallet
237: **for each** $Product\_ID \in S_j$ **do**
238: **if** $CL_i > CL_j$ **then**
239: $CL_j \leftarrow CL_i$
240: **end if**
241: **end for each**

```
242:  end for each
243:  if Product_ID ∈ LP_i = 0 then                ▷ How many Product_ID LP_i are left in the current order
244:      go to 26
245:  end if
246:  if Product_ID ∈ LP_i = 1 then
247:      go to 252
248:  end if
249:  if Product_ID ∈ LP_i > 1 then
250:      go to 267
251:  end if
252:  if CL_i < CL_j then                          ▷ Compare the fragility class of CL_i Product_ID with the fragility class of the pallet unit load CL_j
253:      if ∃_j : Considered_j = false then       ▷ Whether a mix type pallet loading unit was built to which adding Product_IDs was not considered
254:          S_j = BASE
255:          Considered_j = true
256:          go to 50
257:      else
258:          S_j = BASE
259:          j ← j + 1
260:          S_j ∪ Product_ID
261:          go to 74
262:      end if
263:  else
264:      S_j ∪ Product_I D
265:      go to 26
266:  end if
267:  if CL_1 = CL_2 = CL_n = CL_i & CL_i < CL_j then   ▷ Check the fragility class CL_i of the Product_ID item in the order
268:      go to 282
269:  end if
270:  if CL_1 = CL_2 = CL_n = CL_i & CL_i ≥ CL_j then
271:      go to 431
272:  end if
273:  if CL_1 ≠ CL_2 ≠ CL_n ≠ CL_i & CL_i < CL_j then
274:      go to 282
275:  end if
276:  if CL_1 ≠ CL_2 ≠ CL_n ≠ CL_i & CL_i ≥ CL_j then
277:      go to 431
278:  end if
```

279: **if** $CL_1 \neq CL_2 \neq CL_n \neq CL_i \& CL_i \geq CL_j \& CL_n < CL_j$ **then**
280: **go to** 743
281: **end if**
282: **if** $\exists_i : fw_i \in [0.7; 1.0]$ **then** ▷ Are there full-tier Product_IDs ($fw \in [0.7; 1.0]$) among the order items?
283: **go to** 299
284: **else**
285: **if** $LP_i = 1$ **then** ▷ Calculate the number of Product_ID items with full tiers in the order
286: **if** $fw < 0.7$ **then**
287: **go to** 26
288: **end if**
289: **if** $fw \in [0.7; 1.0]$ **then**
290: $S_j \cup LP_i$ ▷ Add Product_IDs to the pallet load unit taking into account the fragility class CL_i
291: **go to** 26
292: **end if**
293: **else**
294: **if** $LP_i > 1$ **then**
295: **go to** 359
296: **end if**
297: **end if**
298: **end if**
299: **if** $\exists_{i,\ldots,i+n;n>1} : Product_ID_i \cup Product\_IDi + n \& fw_i, fw_{i+n} \in [0.7; 1.0]$ **then** ▷ There are two or more Product_IDs in the order that have not been combied before
300: **go to** 304
301: **else**
302: **go to** 330
303: **end if**
304: **if** $fw_i, fw_{i+n} \in [0.7; 1.0]$ **then** ▷ Is it possible to combine two Product_ID items with full layers?
305: **if** $\|H_i - H_{i+n}\| \leq 10$ **then**
306: **if** $fw_i + fw_{i+n} < 0.7$ **then** ▷ Checking the possibility of combining 3-rd Product_ID
307: **if** $\|H_i - H_{i+m}\| \leq 10 \& \|H_{i+n} - H_{i+n}\| \leq 10 \& fw_{i+m} < 0.7$ **then**
308: $Li \cup Product\_ID_{i+m}$ ▷ Add the product to the layer that is created
309: **go to** 305
310: **else**
311: **go to** 305
312: **end if**
313: **else**
314: **if** $fw \in [0.7; 1.0]$ **then**

```
                if    then
                    go to 299
                else
                    S_j \ (Product_IDi, Product_IDi + n, Product_IDi + m        ▷ Discard previously added items
                    go to 299
                end if
            end if
            if  fw > 1  then
                go to 299
            end if
        end if
    else
        go to 299
    end if
end if
if  ∃_{i,i+n} : Product_ID_i ∪ Product_IDi + n  then        ▷ Are there Product_ID items that were not combined
    if  ‖H_i − H_{i+n}‖ ≤ 10  then
        if  fw < 0.7  then
            go to 330
        end if
        if  fw ∈ [0.7; 1.0]  then
            W ← Product_ID_i ∪ Product_ID_{i+n}                                 ▷ A new layer is built
            S_j ⊎ W                                                             ▷ We add a layer to the pallet taking into account the fragility class
            if  Σ_i^I W_i > W^max & Σ_i^I H_i > H^max  then
                S_j \ W
                go to 330
            else
                if  ∃n : Product_ID_n ∈ z  then                                 ▷ Are there any items left in the order?
                    go to 330
                else
                    S_j = BASE
                    go to 26
                end if
            end if
        end if
        if  fw > 1  then
            go to 330
```

```
            end if
        else
            go to 330
        end if
    else
        go to 26
    end if
    if ∃_{i,n} : Product_I D_i ∪ Product_IDn then                ▷ If there are items (Product_IDs) that were not combined into a layer
        if ‖H_i − H_n < 10 then
            if fw_i ∩ fw_n < 0.7 then
                if ∃_i : i + 1 ≠ n$Produc_I D_i ∈ z then          ▷ Is it possible to add another Product_ID from the order?
                    if ‖H_i − H_{i+1} < 10‖&‖H_n − H_{i+1} < 10‖ then
                        if fw_i ∩ fw_n ∩ fwi + 1 < 0.7 then
                            go to 361
                        end if
                        if fw_i ∩ fw_n ∩ fwi + 1 ∈ [0.7; 1.0] then
                            go to 391
                        end if
                        if fw_i ∩ fw_n ∩ fwi + 1 > 1.0 then
                            W ∪ Product_id_{i‖i+1‖n} where fw_i, fw_i, fw_i < 1.0
                            if fW < 0.7 then
                                go to 361
                            end if
                            if fW ∈ [0.7; 1.0] then
                                go to 391
                            end if
                        end if
                    else
                        go to 361
                    end if
                else
                    if fW_top ∈ [0.7; 1.0]&H_top^diff < 10 then
                        S_j = BASE
                    else
                        S_j = TOP
                    end if
                    go to 26
```

```
            end if
        end if
        if fw_i ∩ fw_n ∈ [0.7; 1.0] then
            S_j ∪ Product_ID_i ∪ Product_ID_n
            if Σ_i^I W_i > W^max ‖ Σ_i^I H_i > H^max then
                go to 359
            else
                if ∃n : Product_ID_n ∈ z then                ▷ Are there any items left in the order?
                    go to 359
                else
                    S_j = BASE
                    go to 26
                end if
            end if
        end if
        if fw_i > 1.0 then
            go to 359
        end if
    else
        go to 359
    end if
else                                                         ▷ There are no items (Product_IDs) that were not combined in one layer
    if ∃_{i,n} : Product_ID_i ∪ Product_IDn then
        fW ← ∩fw_i
        if fW > 1 then
            S_j ∪ Product_ID_i ∀_i : Σ_i fw_i ≤ 1
            W_n ← Σ_i W_Product_ID_i                         ▷ The weight of the items to be added
            W_j ← Σ_j W_j                                    ▷ Current weight of the pallet
            W_j^zpk = W^norm − W_j
            if W_n ≤ 0.7 ∗ W_j^zpk then
                S_j ∪ Product_ID_i ∀_i
                S_j = TOP
                go to 26
            end if
            if W_n > 0.7 ∗ W_j^zpk then
                S_j ∪ Product_ID_i ∀_i : W_n ≤ 0.7 ∗ W_j^zpk
                S_j = TOP
```

426: **go to** 26
427: **end if**
428: **end if**
429: **end if**
430: **end if**
431: **if** $\exists_{i,\ldots,i+n;n>1} : Product_I D_i \cup Product\_IDi + n \& fw_i, fw_{i+n} \in [0.7; 1.0]$ **then** ▷ Are in the order any items (Product_IDs) that have not been combined and have full layers?
432: **go to** 436
433: **else**
434: **go to** 625
435: **end if**
436: **if** $\exists_{i,\ldots,i+n;n\geq 1} : Product_I D_i \cup Product\_IDi + n \& fw_i, fw_{i+n} \in [0.7; 1.0]$ **then** ▷ Are in the order at least two items that have not been combined and have full layers?
437: **go to** 441
438: **else**
439: **go to** 481
440: **end if**
441: **if** $\exists_{i,i+n} \| H_i - H_{i+n} \| < 10$ **then**
442: **if** $fw_i \cap fw_{i+n} < 0.7$ **then** ▷ $Product\_ID_i$ and $Product\_ID_{i+n}$ common layer fill factor
443: **if** $\exists Product\_ID_{i,\ldots} \& fw_{i,\ldots} < 0.7$ **then** ▷ Are there any items in the order that do not have full layers?
444: **if** $\forall_{m:m\neq i,m\neq i+n} \| H_m - H_i \| < 10 \& \| H_m - H_i \| < 10$ **then**
445: $W \cup Product\_Id_{i,\ldots}$ ▷ Create a new layer from $Product\_ID_{i,\ldots}$
446: $Sj \cup W$
447: **if** $fW < 0.7$ **then**
448: **go to** 436
449: **end if**
450: **if** $fW \in [0.7; 1.0]$ **then**
451: **if** $\sum_{i=1}^{I} W_i \leq W^{max} \| \sum_{i=1}^{I} H_i \leq H^{max}$ **then**
452: **go to ??**
453: **else**
454: S_j $setminus W$ ▷ Remove from the palette the previously added layer
455: **go to** 436
456: **end if**
457: **end if**
458: **if** $fW > 1$ **then**
459: **go to** 436
460: **end if**

```
            else
                go to 436
            end if
        end if
    end if
    if fw_i ∩ fw_{i+n} ∈ [0.7; 1.0] then
        S_j ∪ (Product_ID_i, Product_ID_{i+n})                    ▷ Add both items to the palette
        if Σ_{i=1}^{I} W_i ≤ W^max & Σ_{i=1}^{I} H_i ≤ H^max then
            go to 436
        else
            S_j \ (Product_ID_i, Product_ID_{i+n})
            go to 436
        end if
    end if
    if fw_i ∩ fw_{i+n} > 1.0 then
        go to 436
    end if
else
    go to 436
end if
if ∃_{i,i+n} ‖H_i − H_{i+n}‖ < 10 then
    W ← ∅
    W ∪ (Product_ID_i, Product_ID_{i+n})
    if fW < 0.7 then
        if ∃Product_ID_{i,...} & fw_{i,...} < 0.7 then            ▷ Are there any items in the order that do not have full layers?
            go to 481
        else
            if ∃_{i,...,i+n;n} : Product_I D_i ∪ Product_IDi + n then   ▷ Are there any other items that have not been linked together?
                if ‖H_i − H_{i+n} < 10 then
                    W ∪ (Product_I D_i, Product_IDi + n)          ▷ The filling degree of the layer cannot exceed 1.0
                    if fW < 0.7 then
                        if ∃_{i,...,i+n;n>1} : Product_I D_i ∪ Product_IDi + n then   ▷ Are in order other items that were not combined?
                            go to 489
                        else
                            H_cmax ← 0
                            for each domain ∈ W
                                j_add ← −1
```

for each $j \in S_j$ **do**
$S_j \cup W_m$
if $H_j \leq H^{max} \& W_j \leq W^{max}$ **then**
if $H_j > H_{cmax}$ **then**
$H_{cmax} \leftarrow H_j$
$j_{add} \leftarrow j$
else
$S_j \setminus W_m$
$j_{add} \leftarrow -1$
end if
else
$S_j \setminus W_m$
$j_{add} \leftarrow -1$
end if
end for each
if $j_{add} \neq -1$ **then**
if $\exists_i : Product\_ID_i \in z \& W_j + W_i \leq W^{max} \& H_j - H_m + H_i$ **then** ▷ Can we add Product_ID from the order to the top layer?
$S_j \cup Product\_ID_i$
end if
$S_j \leftarrow TOP$
go to 26
end if
end for each
for each $m \in W$ **do**
$W \setminus Product\_ID_i \in W \& H_i = max$ ▷ From each set we remove the Product_ID whose height is the highest
go to 495
end for each
end if
end if
if $fw \in [0.7; 1.0]$ **then**
$W \cup (Product_I D_i, Product\_IDi + n)$
if $W_j \leq W^{max}$ **then**
$\S_j \leftarrow BASE$
go to 26
else
while $W_j + W_{Product\_Idi} \leq W^{max}$ **do** ▷ Add additional Product_id items that do not exceed the maximum weight limit W^{max}

$S_j \cup Product\_ID_i$
end while
if $fw < 0.7$ **then**
go to 481
end if
if $fw \in [0.7; 1.0]$ **then**
$S_j \leftarrow BASE$
go to 26
end if
end if
end if
else
for each $m \in W$ **do** ▷ Add Product_ID to the linked items so as not to exceed the fill factor value $1(fw \leq 1)$
while $fw_i + fW < 1.0$ **do**
$W_m \cup Product\_ID_i$
end while
$S_j \cup W_m$
if $\sum_{i=1}^{I} W_i \leq W^{max} \& \sum_{i=1}^{I} H_i \leq H^{max}$ **then**
$S_j \leftarrow TOP$
go to 26
else
while $\sum_{i=1}^{I} W_i > W^{max} \& \sum_{i=1}^{I} H_i > H^{max}$ **do** ▷ Remove the added Product_id until the W^{max} and H^{max} values of the pallet are not exceeded
$S_j \setminus Product\_ID_i$
end while
$S_j \leftarrow TOP$
go to 26
end if
end for each
end if
else ▷ Add Product_ID for which the height difference exceeds $10[mm]$
if $\sum_{i=1}^{I} W_i \leq W^{max} \& \sum_{i=1}^{I} H_i \leq H^{max}$ **then**
$S_j = TOP$
go to 26
else
while $W_j + W_{Product\_ID_i} \leq W^{max} \& W_j + W_{Product\_ID_i} \leq W^{max}$ **do**
$S_j \cup Product\_ID$

```
                    end while
                    S_j = TOP
                    go to 26
                end if
            end if
        end if
    end if
    if fW ∈ [0.7; 1.0] then
        S_j ∪ Product_ID
        if Σ_{i=1}^{I} W_i ≤ W^max & Σ_{i=1}^{I} H_i ≤ H^max then
            if ∃_{i,n} : Product_ID_i, Product_ID_{i+n} ∈ z & Product_ID_i ⊔ Product_ID_{i+n} then
                go to 481
            else
                §_j ← TOP
                go to 26
            end if
        else
            while W_j + W_{Product_ID_i} ≤ W^max & W_j + W_{Product_ID_i} ≤ W^max do        ▷ Add Product_ID to the palette until W^max or H^max is reached
                S_j ∪ Product_ID_i
            end while
            if fw < 0.7 then
                S_j ← TOP
                go to 26
            end if
            if fw ∈ [0.7; 1.0] then
                if h_diff ≤ 10 then
                    S_j ← BASE
                else
                    S_j ← TOP
                end if
                go to 26
            end if
            if fw > 1.0 then
                while fw > 1.0 do
                    S_j \ Product_ID_i
                end while
                if fw < 0.7 then
```

```
                    S_j ← TOP
                end if
                if fw ∈ [0.7; 1.0] then
                    S_j ← BASE
                end if
                go to 26
            end if
        end if
    end if
else
    while ∑_{i=1}^{I} W_i ≤ W^max & ∑_{i=1}^{I} H_i ≤ H^max & fw ≤ 1 do
        if ∃_n Product_ID_n : H_n = max & ∑_{i=1}^{I} W_i + W_n ≤ W^max & ∑_{i=1}^{I} H_i + H_n ≤ H^max & fw ∩ fw_n ≤ 1 then
            S_j ∑ Product_ID_n
            S_j = TOP
            go to 26
        end if
    end while
end if
if ∃_{i,...,i+n;n≥1} : ‖H_i − H_i ... n‖ ≤ 10 then          ▷ Are there Product_ID items whose height difference does not exceed 10[mm]
    if fw < 0.7 then
        if ∃_i : Product_ID_i ∈ z then                      ▷ There are items in the order that can be added to the pallet unit load
            go to 625
        else                                                ▷ There are no additional items in the order that can be added to the pallet unit load
            S_j ∪ Product_ID_{i...n}                        ▷ Add those that meet the height difference condition
            if ∑_{i=1}^{I} W_i ≤ W^max ‖ ∑_{i=1}^{I} H_i ≤ H^max then
                S_j = TOP
                go to 26
            else
                S_j \ Product_ID_{i...n}                    ▷ Remove from the pallet loading unit previously added items
                if ∃_i : Product_ID_i ∈ z then
                    go to ??
                else
                    go to 26
                end if
            end if
        end if
    end if
```

644: **if** $fw \in [0.7; 1.0]$ **then**
645: $S_j \cup Product\_ID_{i...n}$
646: **if** $\sum_{i=1}^{I} W_i \leq W^{max} \| \sum_{i=1}^{I} H_i \leq H^{max}$ **then**
647: **if** $\exists_i : Product\_ID_i \in z$ **then**
648: **go to** 625
649: **else**
650: $S_j = BASE$
651: **go to** 26
652: **end if**
653: **else**
654: $S_j \setminus Product\_ID_{i...n}$ ▷ Remove from the pallet loading unit previously added items
655: **if** $\exists_i : Product\_ID_i \in z$ **then**
656: **go to** 625
657: **else**
658: **go to** 26
659: **end if**
660: **end if**
661: **end if**
662: **if** $fw > 1.0$ **then**
663: **if** $\exists_i : Product\_ID_i \in z$ **then**
664: **go to** 625
665: **else**
666: **if** $\sum_{i=1}^{I} W_i > W^{max}$ **then**
667: **while** $\sum_{i=1}^{I} W_i > W^{max}$ **do**
668: $S_j \setminus Product\_ID_i : W_i = max$ ▷ Remove the Product_ID of the maximum weight from the pallet unit load until the W^{max} value is reached
669: **end while**
670: **end if**
671: **if** $\sum_{i=1}^{I} H_i > H^{max}$ **then**
672: **while** $\sum_{i=1}^{I} H_i > H^{max}$ **do**
673: $S_j \setminus Product\_ID_i : H_i = max$ ▷ Remove the Product_ID of the maximum height from the pallet unit load until the H^{max} value is reached
674: **end while**
675: **end if**
676: **if** $fw > 1.0$ **then**
677: **while** $fw > 1.0$ **do**
678: $S_j \setminus Product\_ID_i : fw = max$ ▷ Remove the Product_ID of the maximum height from the pallet unit load until the fw value of top

layer is reached

end while
end if
if $\forall_{i,j} : Product\_ID_i, Product\_ID_j \in Top_layer \& \|H_i - H_j\| \leq 10 \& fw \in [0.7; 1.0]$ **then**
$S_j = BASE$
go to 26
end if
if $\forall_{i,j} : Product\_ID_i, Product\_ID_j \in Top_layer \& \|H_i - H_j\| > 10 \& fw < 0.7$ **then**
$S_j = TOP$
go to 26
end if
end if
end if
else
if $fw < 0.7$ **then**
$S_j \cup Product\_ID_{i...n}$
if $\sum_{i=1}^{I} W_i > W^{max}$ **then** ▷ Check whether the weight of the pallet loading unit has not been exceeded
while $\sum_{i=1}^{I} W_i > W^{max}$ **do** ▷ Remove those with the lowest weight until the weight of the pallet loading unit W^{max} is reached
$S_j \setminus Product\_ID_i : W_i = min$
end while
end if
if $\sum_{i=1}^{I} H_i > H^{max}$ **then** ▷ Check whether the weight of the pallet loading unit has not been exceeded
while $\sum_{i=1}^{I} H_i > H^{max}$ **do** ▷ Remove those with the lowest weight until the weight of the pallet loading unit W^{max} is reached
$S_j \setminus Product\_ID_i : H_i = min$
end while
end if
$S_j = TOP$
go to 26
end if
if $fw \in [0.7; 1.0]$ **then**
$S_j \cup Product\_ID_{i...n}$
if $\sum_{i=1}^{I} W_i > W^{max}$ **then** ▷ Check whether the weight of the pallet loading unit has not been exceeded
while $\sum_{i=1}^{I} W_i > W^{max}$ **do** ▷ Remove those with the lowest weight until the weight of the pallet loading unit H^{max} is reached
$S_j \setminus Product\_ID_i : W_i = min$
end while
end if
if $\sum_{i=1}^{I} H_i > H^{max}$ **then** ▷ Check whether the weight of the pallet loading unit has not been exceeded

```
            while ∑_{i=1}^{I} H_i > H^max do                ▷ Remove those with the lowest weight until the weight of the pallet loading unit H^max is reached
                S_j \ Product_ID_i : H_i = min
            end while
        end if
        S_j = TOP
        go to 26
    end if
    if fw > 1.0 then
        S_j ∪ Product_ID_{i...n} : fw_i = max               ▷ Add to the pallet loading unit the Product_ID with the highest degree of layer filling fw_i
        if ∑_{i=1}^{I} W_i > W^max then                     ▷ Check whether the weight of the pallet loading unit has not been exceeded
            while ∑_{i=1}^{I} W_i > W^max do                ▷ Remove those with the lowest weight until the weight of the pallet loading unit H^max is reached
                S_j \ Product_ID_i : W_i = min
            end while
        end if
        if ∑_{i=1}^{I} H_i > H^max then                     ▷ Check whether the weight of the pallet loading unit has not been exceeded
            while ∑_{i=1}^{I} H_i > H^max do                ▷ Remove those with the lowest weight until the weight of the pallet loading unit H^max is reached
                S_j \ Product_ID_i : H_i = min
            end while
        end if
    end if
    if fw ∈ [0.7; 1.0] then
        S_j = BASE
        go to 26
    else
        S_j = TOP
        go to 26
    end if
end if
for each Product_Id_i do
    if CL_i < CL_j then
        go to 282
    else
        go to 431
    end if
end for each
end for each
STOP
```