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) 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 fragility (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 factor 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 degree of fragility 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.

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 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. Updates the palette parameters. The goods are marked as assigned to the pallet - they become planned.

- $\backslash$ - 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: $Set\,S\,of\,pallet\,labeled\,as\,,,TOP''or\,,,BASE''$

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?

31:             **go to** 74
32:         **else**
33:             **for each** $j \in$ S **do**
34:                 **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
35:                     **go to** 751
36:                 **else**
37:                     $W^{max} \leftarrow 0.99 * W^{max}$ ▷ Changing the values of the W^{max} and H^{max} parameters to achieve improved balance
38:                     $H^{max} \leftarrow 0.99 * H^{max}$
39:                     **go to** 13
40:                 **end if**
41:             **end for each**
42:         **end if**
43:     **end if**
44:     **for each** $j \in$ S **do**   ▷ Is it possible to add new $Product\_ID$ to already created pallet loading units?
45:         $W_j^{mix} \leftarrow \frac{\sum_{j=1}^{J} W_j}{W_{max}}$ ▷ Calculate the weight coefficient W_{mix} and the height H_{mix} of a pallet mix loading unit.
46:         $H_j^{mix} \leftarrow \frac{\sum_{j=1}^{J} H_j}{H_{max}}$
47:     **end for each**
48:     $W^{max} \leftarrow max_j(W_j^{mix})$ ▷ Choose a pallet with the maximum weight or height
49:     $H^{max} \leftarrow max_j(H_j^{mix})$
50:     **for each** $j \in$ S **do**
51:         $W_j^{diff} \leftarrow W^{max} - W_j$ ▷ Calculate the difference in the maximum weight and height of the pallet loading unit j and the maximum weight and height, respectively
52:         $H_j^{diff} \leftarrow H^{max} - H_j$
53:     **end for each**
54:     **for each** $i \in$ z **do**   ▷ Select $Product\_ID$ with a weight or height not greater than the differences calculated above
55:         **if** $W_i <= W^{diff} \| H_i <= H^{diff}$ **then**
56:             **go to** 62
57:         **else**
58:             $S_j \leftarrow,, BASE''$ ▷ The BASE type pallet loading unit was built
59:             **go to** 26
60:         **end if**
61:     **end for each**
62:     **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
63:         $H_j^{diff} = H^{max} - H_j$
64:         $W_j^{diff} = W^{max} - W_j$

6

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 of the BASE type
129:                                          S_j = BASE
130:                                          go to 30
131:                                      end if
132:                                  end if
133:                              end if
134:                          end for each
```

135:                        **else**

136:                        **end if**

137:                    **end if**

138:                **end if**

139:            **else**

140:                **go to** 30   ▷ If the height difference $\|H_i - H_{i+n}\| exceeds 10$[mm].

141:            **end if**

142:        **end for each**

143:        **if** $i : fw_i \in [0.7; 1.0] \& \exists_n CL_i < CL_{i+n}$ **then**   ▷ Can further items from the order be placed on the full-layer i Product_ID item due to the fragility class $(CL_i \le CL_{i+n})$

144:            **if** $fw_i + fw_{i+n} <= 1$ **then**

145:                **if** $\sum_{i=1}^{I} W_i < W^{max} \& \sum_{i=1}^{I} H_i < H^{max}$ **then**

146:                    **if** $fw < 0.7$ **then**

147:                        $S_j \cup Product\_ID(S_j, i)$

148:                        $S_j = TOP$

149:                        **go to** 30

150:                    **end if**

151:                    **if** $fw \in [0.7; 1.0]$ **then**

152:                        $S_j \cup Product\_ID(S_j, i)$

153:                        **if** $Try\_Pillars(S_j) = true$ **then**

154:                            $S_j \leftarrow TOP$

155:                            **go to** 30

156:                      **else**

157:                            $S_j \leftarrow BASE$

158:                            **go to** 30

159:                      **end if**

160:                  **end if**

161:                **end if**

162:            **end if**

163:        **else**

164:            $Make\_new\_Pallet(Product\_ID_i)$

165:            $j \leftarrow j + 1$

166:            **if** $fw_i < 0.7$ **then**

167:                $S_j = TOP$

168:                **go to** 30

169:            **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 ∈ z do
176:            if  ∃_n : ‖H_i − H_{i+n}‖ <= 10 then        ▷ Warunek polaczenia dwoch pozycji w warstwe celem dodania do palety
177:                go to 182
178:            else
179:                go to 30
180:            end if
181:          end for each
182:          W^{sum} ← 0
183:          top_layer ← ∅
184:          fw_{top_layer} ← 0
185:          for each  i ∈ z do
186:            if  ‖H_i − H_{i+n}‖ ¡ 10  then               ▷ Mozliwosc polaczenia
187:                top_layer ← top_layer ∪ i + n
188:                W^{sum} ← W^{sum} + W_{i+n}
189:                fw_{top_layer} ← fw_{top_layer} + fw_{i+w}
190:            end if
191:          end for each
192:          if  W^{sum} + W_j < W^{max}  then
193:            if  fw_{top_layer} < 1 & H_j + H_{i+n} < H^{max} then
194:                Create_Pillars()
195:            end if
196:            if  fw_{top_layer} < 0.7  then
197:                S_j ∪ top_layer
198:                S_j = TOP
199:                go to 175
200:            end if
201:            if  fw ∈ [0.7; 1.0]  then
202:                S_j ∪ top_layer
203:                if  try_pillars(S_j) = true  then
204:                    S_j = TOP
205:                    go to 175                             ▷ Koniec budowy
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                                      ▷ usuwamy do limitu wagi
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
218:           $Create\_Pillars()$
219:        end if
220:        if  $fw_{top\_layer} < 0.7$  then
221:           $S_j \cup top\_layer$
222:           $S_j = TOP$
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
228:              $S_j = TOP$
229:              go to 175                                                        ▷ Koniec budowy
230:           else
231:              $S_j = BASE$
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  $Product\_ID \in LP_i = 0$  then                                           ▷ Sprawdz liczbe pozycji w zamowieniu
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
253:         if  ∃_j : Considered_j = false  then
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
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 ≠ CL_2 ≠ CL_n ≠ CL_i & CL_i ≥ CL_j & CL_n < CL_j  then
280:         go to ??
281:     end if
```

```
282:     if  ∃_i : fw_i ∈ [0.7; 1.0]  then                                                    ▷ Czy istnieje Product_ID o pelnej warstwie
283:         go to 299
284:     else
285:         if  LP_i = 1  then                                                               ▷ Sprawdz liczbe pozycji w zamowieniu
286:             if  fw < 0.7  then
287:                 go to 26
288:             end if
289:             if  fw ∈ [0.7; 1.0]  then
290:                 S_j ∪ LP_i                                                              ▷ zgodnie z sekwencja CL
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  ∃_{i,...,i+n;n>1} : Product_I D_i ∪ Product_IDi + n & fw_i, fw_{i+n} ∈ [0.7; 1.0]  then      ▷ Itnieja co najmniej 2 pozycje w zamowieniu, ktore nie byly laczone
300:         go to ??
301:     else
302:         go to 330
303:     end if
304:     if  fw_i, fw_{i+n} ∈ [0.7; 1.0]  then                                                ▷ Czy istnieje mozliwosc polaczenia dwoch pozycji Product_ID o pelnych warstwach
305:         if  ‖H_i − H_{i+n}‖ ≤ 10  then
306:             if  fw_i + fw_{i+n} < 0.7  then                                              ▷ Sprawdzenie mozliwosci dodania 3-rd towaru
307:                 if  ‖H_i − H_{i+m}‖ ≤ 10 & ‖H_{i+n} − H_{i+n}‖ ≤ 10 & fw_{i+m} < 0.7  then
308:                     Li ∪ Product_ID_{i+m}                                                ▷ Dodaj produkt do nowej warstwy
309:                     go to ??
310:                 else
311:                     go to ??
312:                 end if
313:             else
314:                 if  fw ∈ [0.7; 1.0]  then
315:                     if    then
316:                         go to 299
317:                     else
318:                         S_j \ (Product_IDi, Product_IDi + n, Product_IDi + m
```

13

```
319:                    go to 299
320:                  end if
321:                end if
322:              if  fw > 1  then
323:                  go to 299
324:              end if
325:            end if
326:          else
327:            go to 299
328:          end if
329:        end if
330:      if  ∃_{i,i+n} : Product_I D_i ∪ Product_IDi + n  then          ▷ Czy sa pozycje Product_ID, ktore nie byly laczone
331:        if  ‖H_i − H_{i+n}‖ ≤ 10  then
332:          if  fw < 0.7  then
333:            go to 330
334:          end if
335:          if  fw ∈ [0.7; 1.0]  then
336:            W ← Product_ID_i ∪ Product_ID_{i+n}                      ▷ Tworzymy nowa warstwe
337:            S_j ⊎ W                                                   ▷ Dodajemy warstwe z uwzglednieniem CL
338:            if  ∑_i^I W_i > W^{max} & ∑_i^I H_i > H^{max}  then
339:              S_j \ W
340:              go to 330
341:            else
342:              if  ∃n : Product_ID_n ∈ z  then                        ▷ Czy sa kolejne pozycje w zamowieniu
343:                go to 330
344:              else
345:                S_j = BASE
346:                go to 26
347:              end if
348:            end if
349:          end if
350:          if  fw > 1  then
351:            go to 330
352:          end if
353:        else
354:          go to 330
355:        end if
```

```
356:     else
357:         go to 26
358:     end if
359:     if  ∃_{i,n} : Product_ID_i ∪ Product_IDn  then                                    ▷ If there are items (Product_IDs) that were not combined into a layer
360:         if  ‖H_i − H_n < 10  then
361:             if  fw_i ∩ fw_n < 0.7  then
362:                 if  ∃_i : i + 1 ≠ n\$Produc_ID_i ∈ z  then                            ▷ Czy mozna dodac kolejna pozycje z zamowienia
363:                     if  ‖H_i − H_{i+1} < 10‖&‖H_n − H_{i+1} < 10‖  then
364:                         if  fw_i ∩ fw_n ∩ fwi + 1 < 0.7  then
365:                             go to 361
366:                         end if
367:                         if  fw_i ∩ fw_n ∩ fwi + 1 ∈ [0.7; 1.0]  then
368:                             go to 391
369:                         end if
370:                         if  fw_i ∩ fw_n ∩ fwi + 1 > 1.0  then
371:                             W ∪ Product_id_{i‖i+1‖n} where fw_i, fw_i, fw_i < 1.0
372:                             if  fW < 0.7  then
373:                                 go to 361
374:                             end if
375:                             if  fW ∈ [0.7; 1.0]  then
376:                                 go to 391
377:                             end if
378:                         end if
379:                     else
380:                         go to 361
381:                     end if
382:                 else
383:                     if  fW_{top} ∈ [0.7; 1.0]&H_{top}^{diff} < 10  then
384:                         S_j = BASE
385:                     else
386:                         S_j = TOP
387:                     end if
388:                     go to 26
389:                 end if
390:             end if
391:             if  fw_i ∩ fw_n ∈ [0.7; 1.0]  then
392:                 S_j ∪ Product_ID_i ∪ Product_ID_n
```

15

```
393:            if  ∑ᵢᴵ Wᵢ > Wᵐᵃˣ‖∑ᵢᴵ Hᵢ > Hᵐᵃˣ  then
394:                go to 359
395:            else
396:                if  ∃n : Product_IDₙ ∈ z  then                            ▷ Czy sa kolejne pozycje w zamowieniu
397:                    go to 359
398:                else
399:                    Sⱼ = BASE
400:                    go to 26
401:                end if
402:            end if
403:        end if
404:        if  fwᵢ > 1.0  then
405:            go to 359
406:        end if
407:    else
408:        go to 359
409:    end if
410: else                                                    ▷ There are no items (Product_IDs) that were not combined in one layer
411:    if  ∃ᵢ,ₙ : Product_IDᵢ ∪ Product_IDn  then
412:        fW ← ∩fwᵢ
413:        if  fW > 1  then
414:            Sⱼ ∪ Product_IDᵢ∀ᵢ : ∑_ifwᵢ ≤ 1
415:            Wₙ ← ∑_iW_Product_IDᵢ                                         ▷ the weight of the items to be added
416:            Wⱼ ← ∑ⱼ Wⱼ                                                    ▷ current weight of the pallet
417:            Wⱼᶻᵖᵏ = Wⁿᵒʳᵐ − Wⱼ
418:            if  Wₙ ≤ 0.7 * Wⱼᶻᵖᵏ  then
419:                Sⱼ ∪ Product_IDᵢ∀ᵢ
420:                Sⱼ = TOP
421:                go to 26
422:            end if
423:            if  Wₙ > 0.7 * Wⱼᶻᵖᵏ  then
424:                Sⱼ ∪ Product_IDᵢ∀ᵢ : Wₙ ≤ 0.7 * Wⱼᶻᵖᵏ
425:                Sⱼ = TOP
426:                go to 26
427:            end if
428:        end if
429:    end if
```

```
393:            if  $\sum_i^I W_i > W^{max} \| \sum_i^I H_i > H^{max}$  then
394:                go to 359
395:            else
396:                if  $\exists n : Product\_ID_n \in z$  then                ▷ Czy sa kolejne pozycje w zamowieniu
397:                    go to 359
398:                else
399:                    $S_j = BASE$
400:                    go to 26
401:                end if
402:            end if
403:        end if
404:        if  $fw_i > 1.0$  then
405:            go to 359
406:        end if
407:    else
408:        go to 359
409:    end if
410: else                                            ▷ There are no items (Product_IDs) that were not combined in one layer
411:    if  $\exists_{i,n} : Product\_ID_i \cup Product\_IDn$  then
412:        $fW \leftarrow \cap fw_i$
413:        if  $fW > 1$  then
414:            $S_j \cup Product\_ID_i \forall_i : \sum\_i fw_i \leq 1$
415:            $W_n \leftarrow \sum\_i W\_Product\_ID_i$                      ▷ the weight of the items to be added
416:            $W_j \leftarrow \sum_j W_j$                                   ▷ current weight of the pallet
417:            $W_j^{zpk} = W^{norm} - W_j$
418:            if  $W_n \leq 0.7 * W_j^{zpk}$  then
419:                $S_j \cup Product\_ID_i \forall_i$
420:                $S_j = TOP$
421:                go to 26
422:            end if
423:            if  $W_n > 0.7 * W_j^{zpk}$  then
424:                $S_j \cup Product\_ID_i \forall_i : W_n \leq 0.7 * W_j^{zpk}$
425:                $S_j = TOP$
426:                go to 26
427:            end if
428:        end if
429:    end if
```

430:     **end if**
431:     **if** $\exists_{i,...,i+n;n>1} : Product_I D_i \cup Product\_IDi + n \& fw_i, fw_{i+n} \in [0.7; 1.0]$ **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** $\exists_{i,...,i+n;n\geq1} : Product_I D_i \cup Product\_IDi + n \& fw_i, fw_{i+n} \in [0.7; 1.0]$ **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** $\exists Product\_ID_{i,...} \& fw_{i,...} < 0.7$ **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:                     $W \cup Product\_Id_{i,...}$ ▷ Create a new layer from $Product\_ID_{i,...}$
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**
461:                 **else**
462:                     **go to** 436
463:                 **end if**
464:             **end if**

17

```
465:        end if
466:        if  fw_i ∩ fw_{i+n} ∈ [0.7; 1.0]  then
467:            S_j ∪ (Product_ID_i, Product_ID_{i+n})                                    ▷ Add both items to the palette
468:            if  ∑_{i=1}^{I} W_i ≤ W^{max} & ∑_{i=1}^{I} H_i ≤ H^{max}  then
469:                go to 436
470:            else
471:                S_j \ (Product_ID_i, Product_ID_{i+n})
472:                go to 436
473:            end if
474:        end if
475:        if  fw_i ∩ fw_{i+n} > 1.0  then
476:            go to 436
477:        end if
478:    else
479:        go to 436
480:    end if
481:    if  ∃_{i,i+n} ‖H_i − H_{i+n}‖ < 10  then
482:        W ← ∅
483:        W ∪ (Product_ID_i, Product_ID_{i+n})
484:        if  fW < 0.7  then
485:            if  ∃Product_ID_{i,...} & fw_{i,...} < 0.7 then                            ▷ Are there any items in the order that do not have full layers?
486:                go to 481
487:            else
488:                if  ∃_{i,...,i+n;n} : Product_I D_i ∪ Product_ID i + n  then           ▷ Are there any other items that have not been linked together?
489:                    if  ‖H_i − H_{i+n} < 10  then
490:                        W ∪ (Product_I D_i, Product_ID i + n)                          ▷ The filling degree of the layer cannot exceed 1.0
491:                        if  fW < 0.7  then
492:                            if  ∃_{i,...,i+n;n>1} : Product_I D_i ∪ Product_ID i + n  then   ▷ Are in order other items that were not combined?
493:                                go to 489
494:                            else
495:                                H_{cmax} ← 0
496:                                for each  do m ∈ W
497:                                    j_{add} ← −1
498:                                    for each  j ∈ S_j  do
499:                                        S_j ∪ W_m
500:                                        if  H_j ≤ H^{max} & W_j ≤ W^{max}  then
501:                                            if  H_j > H_{cmax}  then
```

502:               $H_{cmax} \leftarrow H_j$

503:               $j_{add} \leftarrow j$

504:             **else**

505:               $S_j \setminus W_m$

506:               $j_{add} \leftarrow -1$

507:             **end if**

508:           **else**

509:             $S_j \setminus W_m$

510:             $j_{add} \leftarrow -1$

511:           **end if**

512:         **end for each**

513:         **if** $j_{add} \neq -1$ **then**

514:           **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?

515:             $S_j \cup Product\_ID_i$

516:           **end if**

517:           $S_j \leftarrow TOP$

518:           **go to** 26

519:         **end if**

520:       **end for each**

521:       **for each** $m \in W$ **do**

522:         $W \setminus Product\_ID_i \in W \& H_i = max$ ▷ From each set we remove the Product_ID whose height is the highest

523:         **go to** 495

524:       **end for each**

525:     **end if**

526:   **end if**

527:   **if** $fw \in [0.7; 1.0]$ **then**

528:     $W \cup (Product_I D_i, Product\_IDi + n)$

529:     **if** $W_j \leq W^{max}$ **then**

530:       $\S_j \leftarrow BASE$

531:       **go to** 26

532:     **else**

533:       **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}

534:         $S_j \cup Product\_ID_i$

535:       **end while**

536:       **if** $fw < 0.7$ **then**

537:         **go to** 481

```
538:                              end if
539:                              if  fw ∈ [0.7; 1.0]  then
540:                                  S_j ← BASE
541:                                  go to 26
542:                              end if
543:                          end if
544:                      end if
545:                  else
546:                      for each  m ∈ W do                          ▷ Add Product_ID to the linked items so as not to exceed the fill factor value 1(fw ≤ 1)
547:                          while  fw_i + fW < 1.0  do
548:                              W_m ∪ Product_ID_i
549:                          end while
550:                          S_j ∪ W_m
551:                          if  ∑_{i=1}^{I} W_i ≤ W^{max} & ∑_{i=1}^{I} H_i ≤ H^{max}  then
552:                              S_j ← TOP
553:                              go to 26
554:                          else
555:                              while  ∑_{i=1}^{I} W_i > W^{max} & ∑_{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
556:                                  S_j \ Product_ID_i
557:                              end while
558:                              S_j ← TOP
559:                              go to 26
560:                          end if
561:                      end for each
562:                  end if
563:              else                                                ▷ We add Product_ID for which the height difference exceeds 10[mm]
564:                  if  ∑_{i=1}^{I} W_i ≤ W^{max} & ∑_{i=1}^{I} H_i ≤ H^{max}  then
565:                      S_j = TOP
566:                      go to 26
567:                  else
568:                      while  W_j + W_{Product\_ID_i} ≤ W^{max} & W_j + W_{Product\_ID_i} ≤ W^{max}  do
569:                          S_j ∪ Product_ID
570:                      end while
571:                      S_j = TOP
572:                      go to 26
573:                  end if
```

```
574:                    end if
575:                end if
576:            end if
577:        if  $fW \in [0.7; 1.0]$  then
578:            $S_j \cup Product\_ID$
579:            if  $\sum_{i=1}^{I} W_i \leq W^{max} \& \sum_{i=1}^{I} H_i \leq H^{max}$  then
580:                if  $\exists_{i,n} : Product\_ID_i, Product\_ID_{i+n} \in z \& Product\_ID_i \sqcup Product\_ID_{i+n}$  then
581:                    go to 481
582:                else
583:                    $\S_j \leftarrow TOP$
584:                    go to 26
585:                end if
586:            else
587:                while  $W_j + W_{Product\_ID_i} \leq W^{max} \& W_j + W_{Product\_ID_i} \leq W^{max}$  do          ▷ Add Product_ID to the palette until $W^{max}$ or $H^{max}$ is reached
588:                    $S_j \cup Product\_ID_i$
589:                end while
590:                if  $fw < 0.7$  then
591:                    $S_j \leftarrow TOP$
592:                    go to 26
593:                end if
594:                if  $fw \in [0.7; 1.0]$  then
595:                    if  $h_{diff} \leq 10$  then
596:                        $S_j \leftarrow BASE$
597:                    else
598:                        $S_j \leftarrow TOP$
599:                    end if
600:                    go to 26
601:                end if
602:                if  $fw > 1.0$  then
603:                    while  $fw > 1.0$  do
604:                        $S_j \setminus Product\_ID_i$
605:                    end while
606:                    if  $fw < 0.7$  then
607:                        $S_j \leftarrow TOP$
608:                    end if
609:                    if  $fw \in [0.7; 1.0]$  then
610:                        $S_j \leftarrow BASE$
```

611:                    **end if**
612:                        **go to** 26
613:                    **end if**
614:                **end if**
615:            **end if**
616:        **else**
617:            **while** $\sum_{i=1}^{I} W_i \leq W^{max} \& \sum_{i=1}^{I} H_i \leq H^{max} \& fw \leq 1$ **do**
618:                **if** $\exists_n Product\_ID_n : H_n = max \& \sum_{i=1}^{I} W_i + W_n \leq W^{max} \& \sum_{i=1}^{I} H_i + H_n \leq H^{max} \& fw \cap fw_n \leq 1$ **then**
619:                    $S_j \sum Product\_ID_n$
620:                    $S_j = TOP$
621:                    **go to** 26
622:                **end if**
623:            **end while**
624:        **end if**
625:        **if** $\exists_{i,\dots,i+n;n \geq 1} : \|H_i - H\_i \dots n\| \leq 10$ **then**   ▷ Are there Product_ID items whose height difference does not exceed 10[mm]
626:            **if** $fw < 0.7$ **then**
627:                **if** $\exists_i : Product\_ID_i \in z$ **then**   ▷ There are items in the order that can be added to the pallet unit load
628:                    **go to** 625
629:                **else**   ▷ There are no additional items in the order that can be added to the pallet unit load
630:                    $S_j \cup Product\_ID_{i \dots n}$ ▷ We add those that meet the height difference condition
631:                    **if** $\sum_{i=1}^{I} W_i \leq W^{max} \| \sum_{i=1}^{I} H_i \leq H^{max}$ **then**
632:                        $S_j = TOP$
633:                        **go to** 26
634:                    **else**
635:                        $S_j \setminus Product\_ID_{i \dots n}$ ▷ Remove from the pallet loading unit previously added items
636:                        **if** $\exists_i : Product\_ID_i \in z$ **then**
637:                            **go to** ??
638:                        **else**
639:                            **go to** 26
640:                        **end if**
641:                    **end if**
642:                **end if**
643:            **end if**
644:            **if** $fw \in [0.7; 1.0]$ **then**
645:                $S_j \cup Product\_ID_{i \dots n}$
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

679:          **end while**

680:        **end if**

681:        **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**

682:　　　　　　　　　$S_j = BASE$
683:　　　　　　　　　**go to** 26
684:　　　　　　　　**end if**
685:　　　　　　　　**if** $\forall_{i,j} : Product\_ID_i, Product\_ID_j \in Top_layer \& \|H_i - H_j\| > 10 \& fw < 0.7$ **then**
686:　　　　　　　　　$S_j = TOP$
687:　　　　　　　　　**go to** 26
688:　　　　　　　　**end if**
689:　　　　　　　**end if**
690:　　　　　　**end if**
691:　　　**else**
692:　　　　　**if** $fw < 0.7$ **then**
693:　　　　　　$S_j \cup Product\_ID_{i...n}$
694:　　　　　　**if** $\sum_{i=1}^{I} W_i > W^{max}$ **then**　　　　　　　　　　　　　　　▷ We check whether the weight of the pallet loading unit has not been exceeded
695:　　　　　　　**while** $\sum_{i=1}^{I} W_i > W^{max}$ **do**　　　▷ We remove those with the lowest weight until the weight of the pallet loading unit W^{max} is reached
696:　　　　　　　　$S_j \setminus Product\_ID_i : W_i = min$
697:　　　　　　　**end while**
698:　　　　　　**end if**
699:　　　　　　**if** $\sum_{i=1}^{I} H_i > H^{max}$ **then**　　　　　　　　　　　　　　　▷ We check whether the weight of the pallet loading unit has not been exceeded
700:　　　　　　　**while** $\sum_{i=1}^{I} H_i > H^{max}$ **do**　　　▷ We remove those with the lowest weight until the weight of the pallet loading unit W^{max} is reached
701:　　　　　　　　$S_j \setminus Product\_ID_i : H_i = min$
702:　　　　　　　**end while**
703:　　　　　　**end if**
704:　　　　　　$S_j = TOP$
705:　　　　　　**go to** 26
706:　　　　　**end if**
707:　　　　　**if** $fw \in [0.7; 1.0]$ **then**
708:　　　　　　$S_j \cup Product\_ID_{i...n}$
709:　　　　　　**if** $\sum_{i=1}^{I} W_i > W^{max}$ **then**　　　　　　　　　　　　　　　▷ We check whether the weight of the pallet loading unit has not been exceeded
710:　　　　　　　**while** $\sum_{i=1}^{I} W_i > W^{max}$ **do**　　　▷ We remove those with the lowest weight until the weight of the pallet loading unit H^{max} is reached
711:　　　　　　　　$S_j \setminus Product\_ID_i : W_i = min$
712:　　　　　　　**end while**
713:　　　　　　**end if**
714:　　　　　　**if** $\sum_{i=1}^{I} H_i > H^{max}$ **then**　　　　　　　　　　　　　　　▷ We check whether the weight of the pallet loading unit has not been exceeded
715:　　　　　　　**while** $\sum_{i=1}^{I} H_i > H^{max}$ **do**　　　▷ We remove those with the lowest weight until the weight of the pallet loading unit H^{max} is reached
716:　　　　　　　　$S_j \setminus Product\_ID_i : H_i = min$
717:　　　　　　　**end while**
718:　　　　　　**end if**

```
719:                    S_j = TOP
720:                    go to 26
721:               end if
722:               if  fw > 1.0  then
723:                    S_j ∪ Product_ID_{i...n} : fw_i = max          ▷ We add to the pallet loading unit the Product_ID with the highest degree of layer filling fw_i
724:                    if  ∑_{i=1}^{I} W_i > W^{max}  then                              ▷ We check whether the weight of the pallet loading unit has not been exceeded
725:                         while  ∑_{i=1}^{I} W_i > W^{max}  do          ▷ We remove those with the lowest weight until the weight of the pallet loading unit H^{max} is reached
726:                              S_j \ Product_ID_i : W_i = min
727:                         end while
728:                    end if
729:                    if  ∑_{i=1}^{I} H_i > H^{max}  then                              ▷ We check whether the weight of the pallet loading unit has not been exceeded
730:                         while  ∑_{i=1}^{I} H_i > H^{max}  do          ▷ We remove those with the lowest weight until the weight of the pallet loading unit H^{max} is reached
731:                              S_j \ Product_ID_i : H_i = min
732:                         end while
733:                    end if
734:               end if
735:               if  fw ∈ [0.7; 1.0]  then
736:                    S_j = BASE
737:                    go to 26
738:               else
739:                    S_j = TOP
740:                    go to 26
741:               end if
742:          end if
743:          for each  Product_Id_i  do
744:               if  CL_i < CL_j  then
745:                    go to 282
746:               else
747:                    go to 431
748:               end if
749:          end for each
750: end for each
751: STOP
```