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 collections of orders. The assignment of goods to a pallet loading unit is made within a single order.

For each ordered product (Product_ID), the number of full layers on the pallet loading unit and the size of the residual layer are calculated.

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.
- $\sqcup$ - 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.
- $\setminus$ - Removing the last added layer or Product_ID from the palette that is currently being built.
- $\cap$ - Determining the layer filling degree 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**
4: **for each** $i \in z$ **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: $Added_i \leftarrow false$
9: $l_i = l_i^{INT} + l_i^R$
10: $fw_i^{INT} \leftarrow 0$ ▷ resztkowy spolczynnik wypelnienia warstwy pelnej
11: $fw_i^R \leftarrow 1 - l_i^R$ ▷ resztkowy wspolczynnik wypelnienia warstwy niepelnej
12: **end for each**
13: **for each** $i \in$z **do**
14: **if** $fw_i^{INT} = 0$ **and** $fw_i^R = 0$ **then**
15: $S \leftarrow S \cup Make\_full\_layers\_pallet(i, CL(i), H^{max}, W^{max}, j)$ ▷ Tworzenie palet z warstw pelnych
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: $sort(i, CL(i), 1)$ ▷ Sortowanie rosnaco po klasie kruchosci
24: $sort(i, CL(i), w_i, -1)$ ▷ W obrebie kazdej klasy kruchosci sortowanie malejaco po liczbie warstw w zamowieniu li
25: **end for each**
26: **for each** $i \in$z **do**
27: **if** $\exists\, i : l_i^{Planned}(:) = false \& l_i^{Planned}(:) = true$ **then**
28: **go to** 50
29: **else**
30: **if** $\exists_i : l_i^{Planned}(1 :) = false$ **then**
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**
35: **go to** 751
36: **else**
37: $W^{max} \leftarrow 0.99 * W^{max}$
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**
45: $W_j^{mix} = \frac{\sum_{j=1}^{J} W_j}{W_{max}}$
46: $H_j^{mix} = \frac{\sum_{j=1}^{J} H_j}{H_{max}}$
47: **end for each**
48: $W^{max} = max_j(W_j^{mix})$
49: $H^{max} = max_j(H_j^{mix})$
50: **for each** $j \in$S **do**
51: $W_j^{diff} = W^{max} - W_j$
52: $H_j^{diff} = H^{max} - H_j$
53: **end for each**
54: **for each** $i \in$z **do**
55: **if** $W_i <= W^{diff} \| H_i <= H^{diff}$ **then**
56: **go to** 62
57: **else**
58: $S_j \leftarrow, , BASE''$ ▷ Pallet is builded
59: **go to** 26
60: **end if**
61: **end for each**
62: **for each** $j \in$S **do**
63: $H_j^{diff} = H^{max} - H_j$
64: $W_j^{diff} = W^{max} - W_j$
65: **for each** $i \in$z **do**
66: **if** $H_i <= H_{diff} \| W_i <= W_{diff}$ **then**
67: **go to ??**
68: **else**

69: $S_j \leftarrow, , BASE''$ ▷ Pallet is builded
70: **go to** 26
71: **end if**
72: **end for each**
73: **end for each**
74: **for each** $i \in$z **do**
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: $count \leftarrow 0$
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**
93: **if** $\exists_n : \|H_i - H_{i+n}\| <= 10$ **then** ▷ Warunek polaczenia dwoch pozycji w warstwe celem dodania do palety
94: **if** $\sum_{k=1}^{I} W_k > W^{max} \| \sum_{k=1}^{I} H_k > H^{max}$ **then**
95: **go to** 74 ▷ Dodanie polaczonych warstw przekroczylo wartosci maksymalne
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** ▷ Stopien wypelnienia warstwy wspolnej
99: **go to** 74
100: **else**
101: **if** $\exists_i : planned_i = false$ **then** ▷ Czy sa kolejne pozycje w zamowieniu
102: **for each** $j \in S$ **do**
103: **if** $W_j + W_i \geq W_{max} \| W_j + W_i \geq W_{max} | CL_i \geq CL_j$ **then** ▷ Mozliwosc dodania do istniejacej palety
104: **if** $fw_j + fw_i \leq 1$ **then**
105: $S_j \cup Product\_ID(S_j, i)$

```
                                    if  fw_i + fw_{i+n} < 0.7  then
                                        i \ i + n
                                        go to 74
                                    else
                                        S_j = BASE
                                        go to 30
                                    end if
                                else
                                    if  fw_i + fw_{i+n} < 0.7  then
                                        i \ i + n
                                        go to 74
                                    end if
                                    if  fw_i + fw_{i+n} ∈ [0.7; 1.0]  then
                                        S_j = BASE
                                        go to 30
                                    end if
                                end if
                            else
                                if  fw_i + fw_{i+n} < 0.7  then
                                    i \ i + n
                                    go to 74
                                else
                                    if  fw_i + fw_{i+n} ∈ [0.7; 1.0]  then
                                        S_j = BASE
                                        go to 30
                                    end if
                                end if
                            end if
                        end for each
                    else
                    end if
                end if
            end if
        else
            go to 30
        end if
    end for each
```

```
143:  if i : fw_i ∈ [0.7; 1.0] & ∃_n CL_i < CL_{i+n} then
144:      if fw_i + fw_{i+n} <= 1 then
145:          if Σ_{i=1}^{I} W_i < W^max & Σ_{i=1}^{I} H_i < H^max then
146:              if fw < 0.7 then
147:                  S_j ∪ Product_ID(S_j, i)
148:                  S_j = TOP
149:                  go to 30
150:              end if
151:              if fw ∈ [0.7; 1.0] then
152:                  S_j ∪ Product_ID(S_j, i)
153:                  if Try_Pillars(S_j) = true then
154:                      S_j ← TOP
155:                      go to 30
156:                  else
157:                      S_j ← 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 ← j + 1
166:      if fw_i < 0.7 then
167:          S_j = TOP
168:          go to 30
169:      end if
170:      if fw_i ∈ [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
```

```
        end if
    end for each
    W^sum ← 0
    top_layer ← ∅
    fw_top_layer ← 0
    for each i ∈ z do
        if ‖H_i − H_{i+n} ¡ 10 then                          ▷ Mozliwosc polaczenia
            top_layer ← top_layer ∪ i + n
            W^sum ← W^sum + W_{i+n}
            fw_top_layer ← fw_top_layer + fw_{i+w}
        end if
    end for each
    if W^sum + W_j < W^max then
        if fw_top_layer < 1&H_j + H_{i+n} < H^max then
            Create_Pillars()
        end if
        if fw_top_layer < 0.7 then
            S_j ∪ top_layer
            S_j = TOP
            go to 175
        end if
        if fw ∈ [0.7; 1.0] then
            S_j ∪ top_layer
            if try_pillars(S_j) = true then
                S_j = TOP
                go to 175                                    ▷ Koniec budowy
            else
                S_j = BASE
                go to 175
            end if
        end if
    else
        while W^sum + W_j > W^max do                         ▷ usuwamy do limitu wagi
            top_layer ← top_layer \ i + n
            W^sum ← W^sum − W_{i+n}
            fw_top_layer ← fw_top_layer − fw_{i+w}
        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 ∪ top_layer
222:                S_j = TOP
223:                go to 175
224:            end if
225:            if fw ∈ [0.7; 1.0] then
226:                S_j ∪ 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 ← 0
237:        for each Product_ID ∈ S_j do
238:            if CL_i > CL_j then
239:                CL_j ← CL_i
240:            end if
241:        end for each
242:    end for each
243:    if Product_ID ∈ 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_ID
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
```

```
            go to 26
        end if
    else
        if LP_i > 1 then
            go to 359
        end if
    end if
end if
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
    go to ??
else
    go to 330
end if
if fw_i, fw_{i+n} ∈ [0.7; 1.0] then                ▷ Czy istnieje mozliwosc polaczenia dwoch pozycji Product_ID o pelnych warstwach
    if ‖H_i − H_{i+n}‖ ≤ 10 then
        if fw_i + fw_{i+n} < 0.7 then                                  ▷ Sprawdzenie mozliwosci dodania 3-rd towaru
            if ‖H_i − H_{i+m}‖ ≤ 10&‖H_{i+n} − H_{i+n}‖ ≤ 10&fw_{i+m} < 0.7 then
                Li ∪ Product_ID_{i+m}                                  ▷ Dodaj produkt do nowej warstwy
                go to ??
            else
                go to ??
            end if
        else
            if fw ∈ [0.7; 1.0] then
                if  then
                    go to 299
                else
                    S_j \ (Product_IDi, Product_IDi + n, Product_IDi + m
                    go to 299
                end if
            end if
            if fw > 1 then
                go to 299
            end if
        end if
    else
        go to 299
```

```
328:        end if
329:    end if
330:    if ∃_{i,i+n} : Product_ID_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 Σ_i^I W_i > W^max & Σ_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** $fw_i \cap fw_n \cap fwi + 1 \in [0.7; 1.0]$ **then**
368: **go to** 391
369: **end if**
370: **if** $fw_i \cap fw_n \cap fwi + 1 > 1.0$ **then**
371: $W \cup Product\_id_{i \| i+1 \| n} where fw_i, fw_i, fw_i < 1.0$
372: **if** $fW < 0.7$ **then**
373: **go to** 361
374: **end if**
375: **if** $fW \in [0.7; 1.0]$ **then**
376: **go to** 391
377: **end if**
378: **end if**
379: **else**
380: **go to** 361
381: **end if**
382: **else**
383: **if** $fW_{top} \in [0.7; 1.0] \& H_{top}^{diff} < 10$ **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** $fw_i \cap fw_n \in [0.7; 1.0]$ **then**
392: $S_j \cup Product\_ID_i \cup Product\_ID_n$
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: $S_j \cup Product\_ID_i \forall_i : \sum\_ifw_i \leq 1$
415: $W_n \leftarrow \sum\_iW\_Product\_ID_i$ ▷ 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** $\exists_{i,\ldots,i+n;n>1} : Product_I D_i \cup Product\_IDi + n \& fw_i, fw_{i+n} \in [0.7; 1.0]$ **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?

```
    go to 441
else
    go to 481
end if
if ∃_{i,i+n} ‖H_i − H_{i+n}‖ < 10 then
    if fw_i ∩ fw_{i+n} < 0.7 then                          ▷ Product_ID_i and Product_ID_{i+n} common layer fill factor
        if ∃Product_ID_{i,...} & fw_{i,...} < 0.7 then     ▷ Are there any items in the order that do not have full layers?
            if ∀_{m:m≠i,m≠i+n} ‖H_m − H_i‖ < 10 & ‖H_m − H_i‖ < 10 then
                W ∪ Product_Id_{i,...}                     ▷ Create a new layer from Product_ID_{i,...}
                Sj ∪ W
                if fW < 0.7 then
                    go to 436
                end if
                if fW ∈ [0.7; 1.0] then
                    if Σ_{i=1}^{I} W_i ≤ W^{max} ‖ Σ_{i=1}^{I} H_i ≤ H^{max} then
                        go to ??
                    else
                        S_j setminus W                     ▷ Remove from the palette the previously added layer
                        go to 436
                    end if
                end if
                if fW > 1 then
                    go to 436
                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 ∈ S_j do
                                        S_j ∪ W_m
                                        if H_j ≤ H^max & W_j ≤ W^max then
                                            if H_j > H_cmax then
                                                H_cmax ← H_j
                                                j_add ← j
                                            else
                                                S_j \ W_m
                                                j_add ← −1
                                            end if
                                        else
                                            S_j \ W_m
                                            j_add ← −1
```

```
                        end if
                    end for each
                    if j_add ≠ −1 then
                        if ∃_i : Product_ID_i ∈ z&W_j + W_i ≤ W^max&H_j − H_m + H_i then ▷ Can we add Product_ID from the order to the top layer?
                            S_j ∪ Product_ID_i
                        end if
                        S_j ← TOP
                        go to 26
                    end if
                end for each
                for each m ∈ W do
                    W \ Product_ID_i ∈ 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 ∈ [0.7; 1.0] then
            W ∪ (Product_ID_i, Product_IDi + n)
            if W_j ≤ W^max then
                §_j ← BASE
                go to 26
            else
                while W_j + W_Product_Idi ≤ W^max do ▷ Add additional Product_id items that do not exceed the maximum weight limit W^max
                    S_j ∪ Product_ID_i
                end while
                if fw < 0.7 then
                    go to 481
                end if
                if fw ∈ [0.7; 1.0] then
                    S_j ← BASE
                    go to 26
                end if
            end if
        end if
    else
        for each m ∈ W do ▷ Add Product_ID to the linked items so as not to exceed the fill factor value 1(fw ≤ 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 ▷ We 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 $fW \in [0.7; 1.0]$ **then**
$S_j \cup Product\_ID$
if $\sum_{i=1}^{I} W_i \leq W^{max} \& \sum_{i=1}^{I} H_i \leq H^{max}$ **then**
if $\exists_{i,n} : Product\_ID_i, Product\_ID_{i+n} \in z \& Product\_ID_i \sqcup Product\_ID_{i+n}$ **then**
go to 481
else

$\S_j \leftarrow TOP$
go to 26
end if
else
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
$S_j \cup Product\_ID_i$
end while
if $fw < 0.7$ **then**
$S_j \leftarrow TOP$
go to 26
end if
if $fw \in [0.7; 1.0]$ **then**
if $h_{diff} \leq 10$ **then**
$S_j \leftarrow BASE$
else
$S_j \leftarrow TOP$
end if
go to 26
end if
if $fw > 1.0$ **then**
while $fw > 1.0$ **do**
$S_j \setminus Product\_ID_i$
end while
if $fw < 0.7$ **then**
$S_j \leftarrow TOP$
end if
if $fw \in [0.7; 1.0]$ **then**
$S_j \leftarrow BASE$
end if
go to 26
end if
end if
end if
else
while $\sum_{i=1}^{I} W_i \leq W^{max} \& \sum_{i=1}^{I} H_i \leq H^{max} \& fw \leq 1$ **do**
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**
$S_j \sum Product\_ID_n$

```
            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}                            ▷ We add those that meet the height difference condition
            if Σ_{i=1}^{I} W_i ≤ W^max ‖ Σ_{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
    if fw ∈ [0.7; 1.0] then
        S_j ∪ Product_ID_{i...n}
        if Σ_{i=1}^{I} W_i ≤ W^max ‖ Σ_{i=1}^{I} H_i ≤ H^max then
            if ∃_i : Product_ID_i ∈ z then
                go to 625
            else
                S_j = BASE
                go to 26
            end if
        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 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: $S_j \cup Product\_ID_{i...n} : fw_i = max$ ▷ We add to the pallet loading unit the Product_ID with the highest degree of layer filling fw_i
724: **if** $\sum_{i=1}^{I} W_i > W^{max}$ **then** ▷ We check whether the weight of the pallet loading unit has not been exceeded
725: **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
726: $S_j \setminus Product\_ID_i : W_i = min$

```
            end while
        end if
        if ∑_{i=1}^{I} H_i > H^max then                ▷ We check whether the weight of the pallet loading unit has not been exceeded
            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
                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
```