BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 69(1), 2021, Article number: e135836
DOI: 10.24425/bpasts.2021.135836

CONTROL AND INFORMATICS

Rack cell configuration problem: a mathematical model and effective combined heuristic

Andrzej RATKIEWICZ* and Konrad LEWCZUK

Warsaw University of Technology, Faculty of Transport, Koszykowa 75, 00-662 Warsaw, Poland

Abstract. This paper discusses the configuration of a space-effective rack cell for storing a given set of heterogeneous items. Rack cells are the primary components of rack storage areas. A rack cell configuration problem (RCCP) for heterogeneous storage is formulated as a combinatorial mathematical model. An effective heuristic for solving the RCCP in practical cases is presented. The proposed heuristic consists of multistage brute force searching of defined sets of feasible solutions and solving linear integer assignment problems by the branch-and-bound method. The developed algorithm was implemented and tested, and the rack cell obtained meets the modularity requirements in the design and operation of heterogeneous storage areas.

Key words: combinatorial optimization; rack cell configuration; space utilisation; storage; cutting and packing problems.

1. Introduction

This work focuses on the optimization of space utilization in storage areas. In the case of uniform items stored in identical pallet racks, the storage area is an easy-to-design modular system. However, in the case of non-uniform items (heterogeneous storage) two antagonistic design approaches may be employed:

1. All items are stored in identical rack cells, resulting in modularity and flexibility, as well as possible significant space wastage, owing to the mismatch between the dimensions of the rack cells and stored items.
2. Each item type is stored in a dedicated rack cell, which ensures improved space utilisation, but also results in a lack of modularity, difficult design, and inflexibility.

Both options exhibit the disadvantages of high investment expenditures and/or poor space utilization. Therefore, wider research within the rack cell configuration problem (RCCP) is necessary to provide a method that serves as a compromise between the two antagonistic options.

The RCCP can be described as follows: the dimensions and weights of all items (palletized units, boxes) to be stored are known, as well as the limits of the dimensions of the rack cells (bins) in which to place items. Then, the number of bins, dimensions of the particular bins, and item arrangements in each bin must be determined. The problem includes the possible rotation of items (space is configured in three dimensions x, y and z, vertical axis rotation in 90 degree increments only; see Fig. 1).

The problem occurs in large distribution and reserve warehouses [1, 2], but also in smaller storage areas, such as retail outlets, job-shops on production lines, and service depots [3]. In general, rack cell dimensions may differ, and depend on the assumptions of the storage or picking area design. Storage space is shaped by means of the racking construction, area dimensions, and pillar grid. Racking construction defines the work aisles for material handling (see [1]) and rack cells for storage. Rack cells are uniform cuboids limited by rack construction elements (Fig. 1).

Cells must be set to carry all unit types in a system, while maintaining the necessary technological safety gaps. For safety reasons, it is also important to keep the verticality of pillars and the levelness of shelves [5]. Grounds arising from the lattice structure of racking systems can use cells of the same length and depth, but with possible height variations. The level of a pair of rack beams can be adjusted up or down to fit the cell to the unit height but changing the pillar arrangements (rack bays) is difficult and requires alteration of the construction elements. Owing to the statics of the rack structure, all rack bays should have the same dimensions and, in particular, the same depth (Fig. 1).

The RCCP encompasses two issues. The first is slotting materials into the fixed shelving or racking system. This can be effectively solved using well-known methods concerning the bin-packing problem. However, this restricts potential applications to cases with determined rack cell dimensions. The second issue is determining which rack cell dimensions are optimal for a given set of items.

2. Background and discussion

Formally, several publications have discussed the problem of rack cell dimensions, but in most cases, these have been taken as constant and known values. The dimensions of rack cells of different types and for various purposes have been assumed as fixed and have not been discussed in a wide range of design problems. However, these dimensions are very often parameters of optimisation models, influencing the final research results [6–8].

*e-mail: andrzej.ratkiewicz@pw.edu.pl

Manuscript submitted 2020-03-16, revised 2020-07-22, initially accepted for publication 2020-09-26, published in February 2021

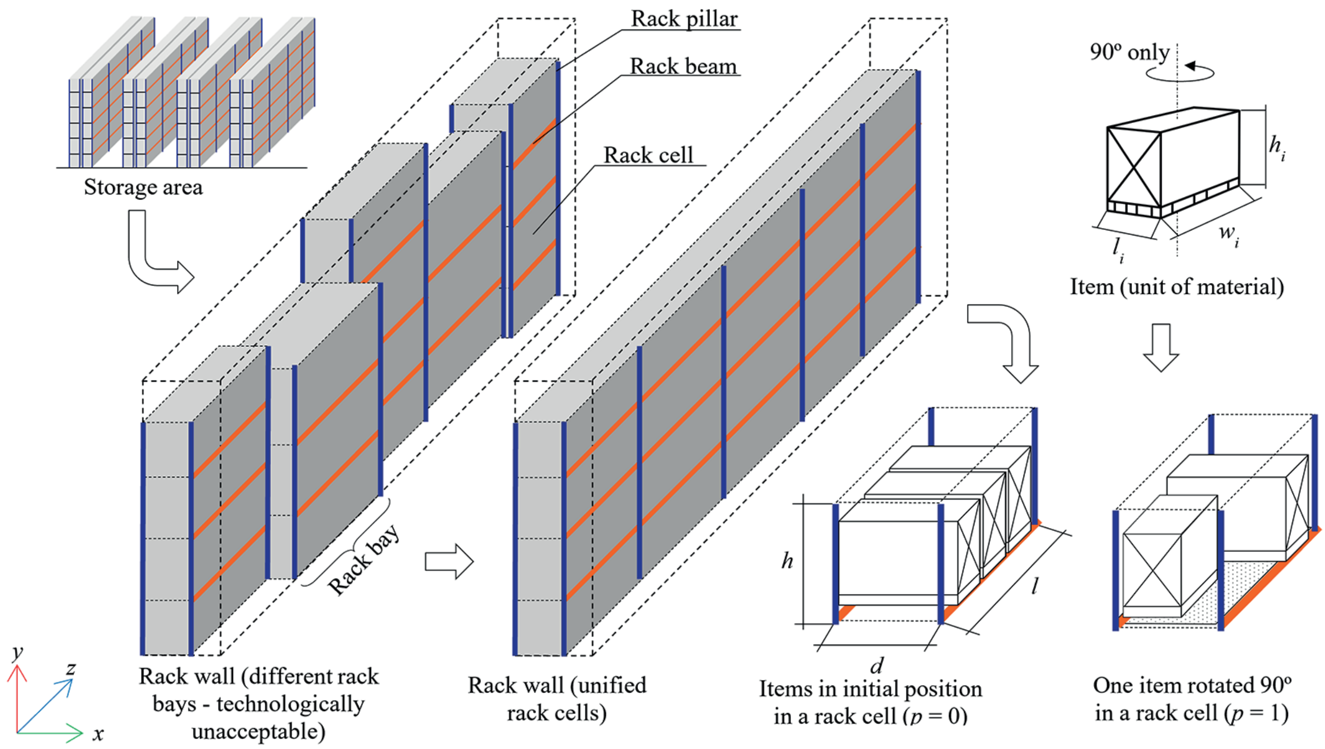

Fig. 1. Racking system scheme for heterogeneous storage

Reference [9] described the AS/RS model with a rack of modular cells, according to the criteria of wasted space and exploitation costs. They provided an exact mathematical model as well as a heuristic method for the problem solution and defined the method of setting the rack cell dimensions for non-uniform material units. The study addressed the same problem but provided less significant results than the solution presented in this paper. The most important differences are that the model formulated below minimises space wastage, not only according to the rack cell and unit heights, but also considering all other dimensions, weights, and possible unit rotations.

Reference [10] presented the problem of storage rack arrangement for non-uniform load units in a storage area, under constraints imposed by carrying pillars of building construction. An exact model for the warehouse building cost minimisation and space consumption minimisation was defined. A bi-level approach containing both a mathematical model and coordinating procedure for the problem solution was presented.

The extended literature review justifies the statement that list of methods for arranging storage areas with concurrent optimisation of the rack cells doesn't exist.

Proper recognition of the RCCP research area is fundamental to the course of studies (Fig. 2). Intuitively, RCCP can be bracketed together with cutting and packing problems (CPPs). Older scientific studies systematising the CPP (as in [11]) did not allow for classification of the RCCP. The closest to the RCCP specificity is the taxonomy presented by Wäscher in [12] and Silva in [13]. Wäscher classified the

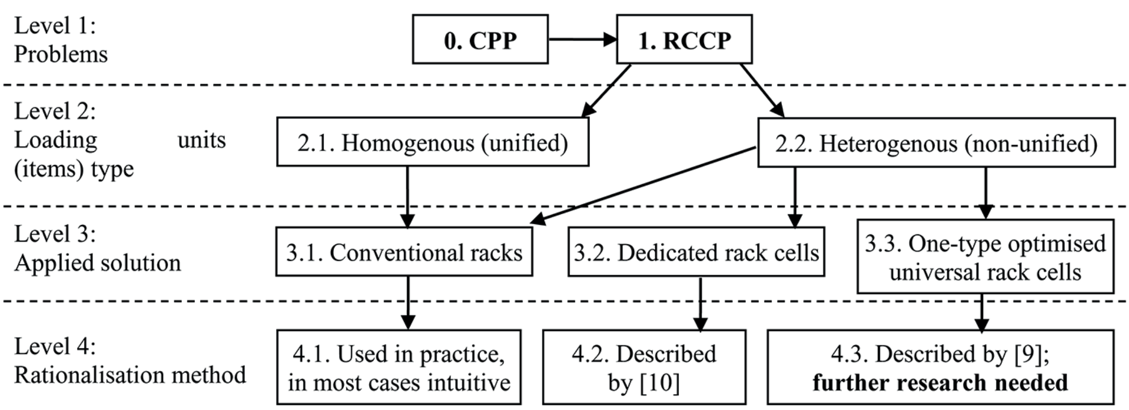

Fig. 2. General systematics of RCCP

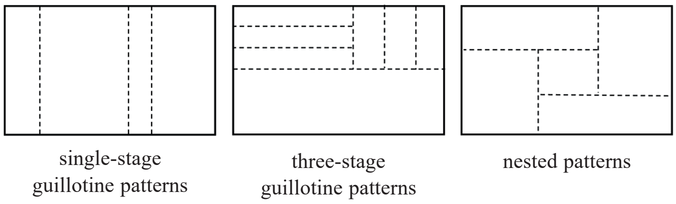

Fig. 3. Examples of orthogonal patterns in CPPs. Based on [11]

arrangement of small items within one large object with variable dimension(s) as the open dimensional problem (see Fig. 11 in Wäscher's publication). In the general form of the RCCP, dimensions of more than one bin must be determined, which is why it defies Wäscher's classification. Silva introduced the concept of the "open dimension problem with a weakly and a strongly heterogeneous assortment ... of the large objects", but they permitted only one dimension to be changed in the large objects (p. 848). Therefore, the general form of the RCCP, in which all dimensions of a large object are set, also does not fit Silva's classification. It should be noted that also newer publications on CPP typology like [14, 15] did not mention RCCP in the context presented in this paper. The following discussion is focused on 4.3 (Fig. 2) of the general systematics of the RCCP.

In order to maintain the technological requirements, items must be arranged in a rack cell up to the single-stage guillotine pattern (Fig. 3), as discussed in [11].

The RCCP configuration method in the presented form is an effective rack design tool providing optimal results (with a globally optimal solution to the equations system using the branch-and-bound) in an acceptable time. Such methods are not present in the literature, except their particular elements – without a comprehensive approach. The problem noted below requires a dedicated heuristic solving algorithm combining elements from various research approaches into new form.

3. Heuristic based solution method

3.1. Problem description. The paper presents the "static" situation of the rack arrangement. The solution to the dynamic version of the problem could be a cyclical application of discussed method to the storage system. However, the limitation of the cyclic use of the proposed approach is the high cost of rearranging the rack area. The problem involves determining the *optimal dimensions of a rack cell to maximise storage space usage for a set of non-uniform items*. Let items of different types exist, with known quantities (volumes) for each type. Different item types have varying dimensions and weights. Using only the standard components of racking systems, such as longitudinal beams and frames, one must set the dimensions of the *universal rack cell* (URC – indicated in 3.3, Fig. 2) to be used in the storage area. The URC must be able to hold all item types and provide direct access to each item (single-stage guillotine pattern). The storage area using these cells must be able to keep a specified number of items at a time. The criterion is the lowest cubic volume of the storage space as a sum of the cubes of the individual cells. The space for working aisles, which is relevant mostly in material handling technology, is not taken into account.

Let the following indexes be defined:

- i – type of item, $\boldsymbol{I} = \{1, 2, \ldots, i, \ldots, I\}$;
- p – variant of item rotation, $p \in \{0, 1\}$, where 1 means the item was rotated from its initial position by 90° (Fig. 1);
- s – type of rack beam, $\boldsymbol{S} = \{1, 2, \ldots, s, \ldots, S\}$;
- n – variant of the arrangement of items in a rack cell, $\boldsymbol{N} = \{1, 2, \ldots, n, \ldots, N\}$; for each feasible configuration of the URC (according to the available beam types, item height, width, weight and rotation, and constraint for maximal cell depth), the possible combinations of item arrangements in that cell can be determined. The number N of arrangement combinations in all feasible rack cells can be estimated (Fig. 4).

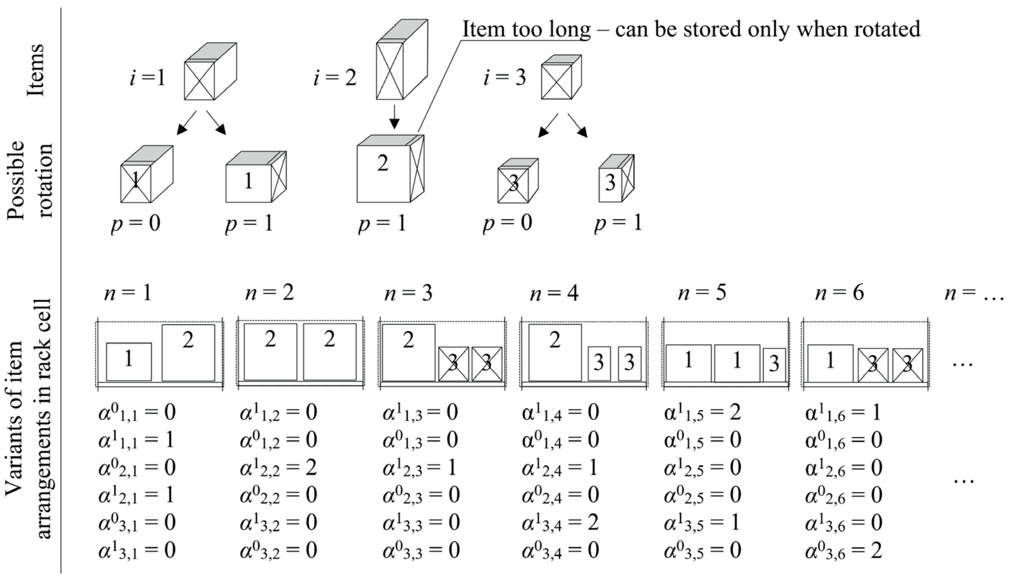

Fig. 4. Interpretation of $\alpha^{p}_{i,n}$ variable

The model parameters are as follows:

- λ_i – number of i-type items;
- w_i, l_i, h_i – initial width, length, and height of i-type item [mm];
- x_i, y_i – absolute width and length of i-type item (when rotated or not) [mm];
- c_i – weight of i-type item [kg];
- d, l, h – depth, length, and height of URC [mm];
- $d^{\max}$ – maximum depth of rack cell (assumed) [mm];
- l_s – length of s-type rack beam [mm];
- g_s – height (thickness) of s-type rack beam [mm];
- c_s – loading capacity of pair of s-type rack beams [mm];
- b_1 – width (thickness) of rack pillar [mm];
- b_2 – vertical safety gap between items or item and rack pillar [mm];
- b_3 – horizontal safety gap (clearance area) between item and rack beam [mm].

The decision variables are:

- κ_s – binary variable determined using s-type rack beam in URC, $\mathbf{K} = [\kappa_s \in \{0, 1\}; s \in \boldsymbol{S}]$;
- $\alpha^p_{i,n}$ – integer variable for a number of i-type items of p-th rotation constituting n-th arrangement variant (Fig. 4), $\mathbf{A} = [\alpha^p_{i,n} \in \boldsymbol{C}^+ \cup \{0\}; p \in \{1, 2\}; i \in \boldsymbol{I}; n \in \boldsymbol{N}]$;
- $\beta_{n(s)}$ – number of rack cells using $n(s)$-th arrangement variant, $\mathbf{B} = [\beta_{n(s)} \in \boldsymbol{C}^+ \cup \{0\}; n \in \boldsymbol{N}_s]$.

When the i-type item is going to be stored rotated by 90° from the initial position, the length l_i and width w_i of this item are replaced by Eq. (1) and Eq. (2). Then, the absolute width and length of the i-type item are marked as x_i and y_i, respectively. The height of the items cannot be changed by any rotation.

$$\forall i \in \boldsymbol{I} \;\; x_i = pw_i + (1-p)l_i, \tag{1}$$

$$\forall i \in \boldsymbol{I} \;\; y_i = pl_i + (1-p)w_i. \tag{2}$$

There must exist a rack beam that is suitable for the length of each item in at least one rotation Eq. (3). If not, the problem does not have a feasible solution.

$$\forall i \in \boldsymbol{I} \;\; \exists s \in \boldsymbol{S} \;\; x_i + 2b_2 \leq l_s \tag{3}$$

Only one type of beam is used in the URC Eq. (4). The cell length is limited by the given series of beam types and includes the rack pillar thickness Eq. (5). The sum of lengths of items assigned to the cell and safety gaps must not exceed the length of the used rack beam Eq. (6).

$$\sum_{s \in \boldsymbol{S}} \kappa_s = 1, \tag{4}$$

$$l = \sum_{s \in \boldsymbol{S}} \kappa_s l_s + b_1, \tag{5}$$

$$\forall n \in \boldsymbol{N} \;\; b_2 + \sum_{i \in \boldsymbol{I}} \sum_{p \in \{0,1\}} \alpha^p_{i,n}(x_i + b_2) \leq \sum_{s \in \boldsymbol{S}} \kappa_s l_s. \tag{6}$$

The URC height is fixed and results from the beam height, highest item, and horizontal safety gap Eq. (7).

$$h = \max_{i \in \boldsymbol{I}} \{h_i\} + \sum_{s \in \boldsymbol{S}} \kappa_s g_s + b_3 \tag{7}$$

The URC depth is imposed by the dimensions and rotation of the assigned items Eq. (8). All items in the cell must be rotated in a manner ensuring that the depth of a cell is not greater than the maximal value Eq. (9). The total weight of items cannot exceed the loading capacity of the rack beams Eq. (10).

$$d = \max_{i \in \boldsymbol{I},\, p \in \{0,1\},\, n \in \boldsymbol{N}} \left\{ y_i \cdot \operatorname{sgn}\left(\alpha^p_{i,n}\right) \right\} \tag{8}$$

$$d \leq d^{\max} \tag{9}$$

$$\forall n \in \boldsymbol{N} \;\; \sum_{i \in \boldsymbol{I}} \sum_{p \in \{0,1\}} \alpha^p_{i,n} c_i \leq \sum_{s \in \boldsymbol{S}} \kappa_s c_s \tag{10}$$

The storage area constructed for the URC must be able to store all items of each type Eq. (11).

$$\forall i \in \boldsymbol{I} \;\; \sum_{p \in \{0,1\}} \sum_{n \in \boldsymbol{N}} \beta_n \alpha^p_{i,n} \geq \lambda_i \tag{11}$$

The primary goal is to minimise the storage area cubic volume Eq. (12), which is the product of the number of URCs and their cubic volumes.

$$\min \sum_{n \in \boldsymbol{N}} \beta_n \cdot lhd \tag{12}$$

The fully expanded criteria function provided by formula (12) takes the following form:

$$\begin{aligned} \min_{\alpha^p_{i,n} \in \boldsymbol{A},\, \beta_n \in \boldsymbol{B},\, \kappa_s \in \boldsymbol{K}} & \sum_{n \in \boldsymbol{N}} \beta_n \left(\sum_{s \in \boldsymbol{S}} \kappa_s l_s + b_1 \right) \left(\max_{i \in \boldsymbol{I}} \{h_i\} + \right. \\ & \left. + \sum_{s \in \boldsymbol{S}} \kappa_s g_s + b_3 \right) \left(\max_{i \in \boldsymbol{I},\, p \in \{0,1\},\, n \in \boldsymbol{N}} \left\{ y_i \cdot \operatorname{sgn}\left(\alpha^p_{i,n}\right) \right\} \right). \end{aligned} \tag{13}$$

The proposed mathematical model is a nonlinear mixed-integer programming problem. As mentioned previously, the RCCP can be associated with CPPs, which are stated as NP-hard problems. This means that no known algorithm exists for solving these problems, such that the computational effort at worst increases as a polynomial in the problem size. Therefore, it is expected that the RCCP will inherit the NP-hardness.

The main difficulty arises from the unknown number of possible variants of the item arrangements in the rack cell. Each feasible configuration of cell dimensions has different allowed combinations of item arrangements (Fig. 4). Determination of the sizes of matrices **A** and **B** requires this number, but it can be only obtained by brute force searching for given sets of items, rack beams, and constraints. This means that full formulation of the proposed optimisation task requires significant initial efforts.

Trivial cases, for which the optimal solution can be delivered intuitively, were optimally solved in negligible time by the LINGO software in order to confirm the formal correctness of the presented model. For non-trivial cases, LINGO did not yield any results within an acceptable time, so an alternative method for solving the RCCP must be developed for practical use.

3.2. Two-stage combined heuristic. A simplified, time-effective, two-stage combined heuristic approach with high chances of optimality was then proposed. During the first stage, all feasible rack cells were identified and a complete review of the item arrangements in each feasible cell was conducted. In the second stage, a batch of simple linear integer assignment problems was tackled using the branch-and-bound method. An effective brute force review on the first stage was enabled by narrowing the space of feasible solutions by simplifying the assumptions based on the technological qualities of a RCCP. The method provides proper and applicable results for large-scale examples within a reasonable time.

The heuristic solution algorithm is detailed in a framed *pseudo-code section* supported by a simple calculation example, as illustrated in Figs. 5 to 9 (the symbols as in Section 3.1).

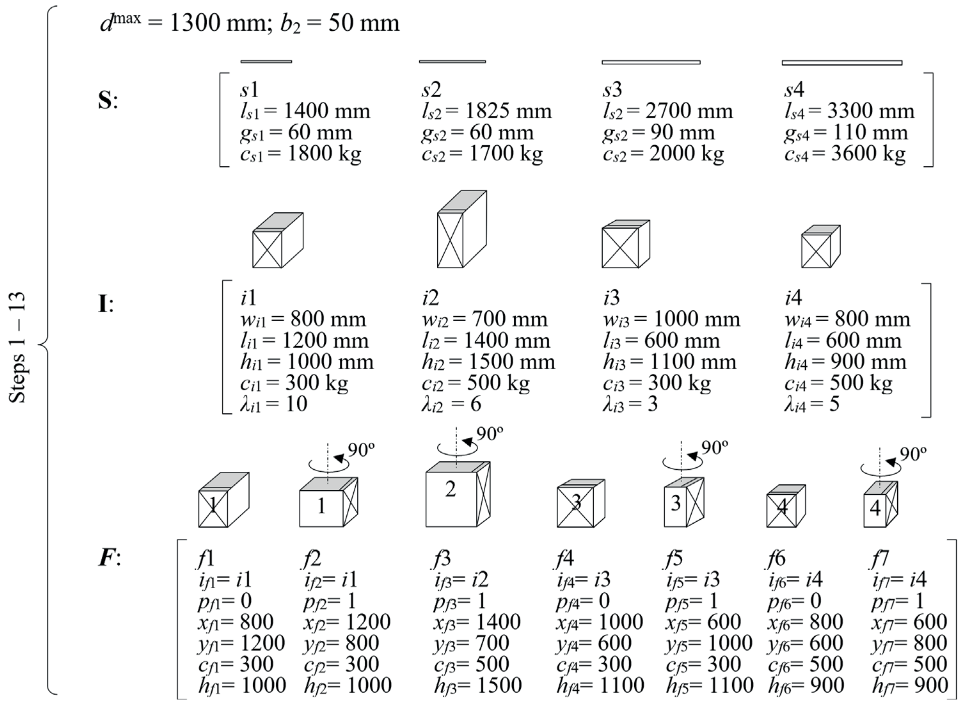

Fig. 5. Visualisation of steps 1 to 13 of solving algorithm

Steps 14 – 18

Check rack beams usability:

assume $a = \max\{\min\{x_{f1},x_{f1}\}, x_{f3}, \min\{x_{f4},x_{f5}\}, \min\{x_{f6},x_{f7}\}\}+2b_2 = 1400+2\cdot 50 = 1500$ mm
assume $b = \max\{c_i\} = 500$ kg, then:

check *s1*: $[(a \le l_{s1})$ or $(b > c_{s1})]$ is not true, remove beam *s*1 from **S**
check *s2*: $[(a \le l_{s2})$ or $(b > c_{s2})]$ is true, beam suitable
check *s3*: $[(a \le l_{s3})$ or $(b > c_{s3})]$ is true, beam suitable
check *s4*: $[(a \le l_{s4})$ or $(b > c_{s4})]$ is true, beam suitable

Steps 19 – 25

Find minimal depth d^{min} of universal rack cell

$d^{min} = \max\{\min\{y_{f1},y_{f2}\}, y_{f3}, \min\{y_{f4},y_{f5}\}, \min\{y_{f6},y_{f7}\}\} =$
$= \max\{\min\{1200, 800\}, 700, \min\{600, 1000\}, \min\{500, 600\}\} = 800$ mm

Steps 26 – 32

Construct set of potential depths $\boldsymbol{D}_s$ of universal rack cell for *s*-type beam (example only for *s*4):

$x_{f1} = 800$ is $\le l_s - 2g_v = 3200$, then $y_{f1} = 1200$ is $\ge d^{min}$, so d^{temp} = <u>1200</u>
$x_{f2} = 1200$ is $\le l_s - 2g_v = 3200$, then $y_{f2} = 800$ is $\ge d^{min}$, so d^{temp} = <u>800</u>
$x_{f3} = 1400$ is $\le l_s - 2g_v = 3200$, then $y_{f3} = 700$ is not $\ge d^{min}$
$x_{f4} = 1000$ is $\le l_s - 2g_v = 3200$, then $y_{f4} = 600$ is not $\ge d^{min}$
$x_{f5} = 600$ is $\le l_s - 2g_v = 3200$, then $y_{f5} = 1000$ is $\ge d^{min}$, so d^{temp} = <u>1000</u>
$x_{f6} = 800$ is $\le l_s - 2g_v = 3200$, then $y_{f6} = 600$ is not $\ge d^{min}$
$x_{f7} = 600$ is $\le l_s - 2g_v = 3200$, then $y_{f7} = 800$ is $\ge d^{min}$, so d^{temp} = 800

$\boldsymbol{D}_{s4}$ = {1200, 1000, 800}

Fig. 6. Visualisation of steps 14 to 32 of solving algorithm

1: Set $d^{\max}$, b_1, b_2, b_3
2: Set matrix **S** of rack beam types, $s = [l_s, g_s, c_s]$
3: Set matrix **I** of item types, $i = [w_i, l_i, h_i, c_i, \lambda_i]$
4: Create matrix $\mathbf{F} = NULL$ of feasible variants of item rotation, $f = [i_f, p_f, x_f, y_f, h_f, c_f]$
Convert matrix **I** into matrix **F** of item rotation variants (Fig. 5):
5: For all i in **I** do
6: If $l_i \le d^{\max}$ then //If item length is less than maximal rack cell depth…
7: create new f //create new variant of item rotation.
8: attribute: $i_f = i$; $p_f = 0$; $x_f = w_i$; $y_f = l_i$; $h_f = h_i$; $c_f = c_i$;
9: include f to **F**
10: If $w_i \le d^{\max}$ and $w_i \ne l_i$ then //If item width is less than maximal rack cell depth…
11: create new f //create new variant of item rotation.
12: attribute: $i_f = i$; $p_f = 1$; $x_f = l_i$; $y_f = w_i$; $h_f = h_i$; $c_f = c_i$
13: include f to **F**
Check all rack beams for potential usability (Fig. 6):
14: For all s in **S** do
15: For all i in **I** do
16: If not $\{[(w_i \le l_s + 2b_2$ and $l_i \le d^{\max})$ or $(l_i \le l_s + 2b_2$ and $w_i \le d^{\max})]$ or $(c_i \ge c_s)\}$
17: remove s from **S** //beam cannot keep all item types
18: If **S** is $NULL$ then go to STOP //no feasible solution.
Set minimal possible depth of universal rack cell $d^{\min}$ (Fig. 6):
19: $d^{\min} = 0$ //set minimal depth of universal rack cell as 0
20: For all i in **I** do
21: $d^{\text{temp}} = \infty$ //set temporary maximal value
22: For all f in **F** do
23: If $i_f = i$ then
24: $d^{\text{temp}} = \min\{d^{\text{temp}}, y_f\}$
25: $d^{\min} = \max\{d^{\min}, d^{\text{temp}}\}$ //find minimal depth of universal rack cell $d^{\min}$
For all beams in matrix **S** construct set of potential rack cell depths $\boldsymbol{D}_s$ (Fig. 6):
26: For all s in **S** do
27: create set $\boldsymbol{D}_s = NULL$ of d_s
//…set of potential depths of URL based on s-th beam
28: For all f in **F** do
29: If $(x_f \le l_s - 2b_2)$ and $(y_f \le d^{\min})$
then //If variant of item rotation fits length of beam and minimal depth of rack cell
30: attribute $d_s = y_f$ //attribute depth of variant as potential depth of rack cell
31: If $d_s \notin \boldsymbol{D}_s$ then //if this depth is still not present in the set of potential depths
32: include d_s to $\boldsymbol{D}_s$

Construct matrix **M** of potential rack cell footprints (Fig. 7):
33: Create matrix $\mathbf{M} = NULL$ of
rack cell footprints $m = [l_m, d_m, s_m]$
34: For all s in **S** do //The footprint is constructed on the base of beam length, and
35: For all d_s in $\boldsymbol{D}_s$ do //…potential depths of rack cells.
36: create new m
36: attribute: $l_m = l_s + b_1$; $d_m = d_s$; $s_m = s$;
38: include m to **M**
Construct matrix $\mathbf{N}_m$ of feasible rack cells for potential rack cell footprints (Fig. 7):

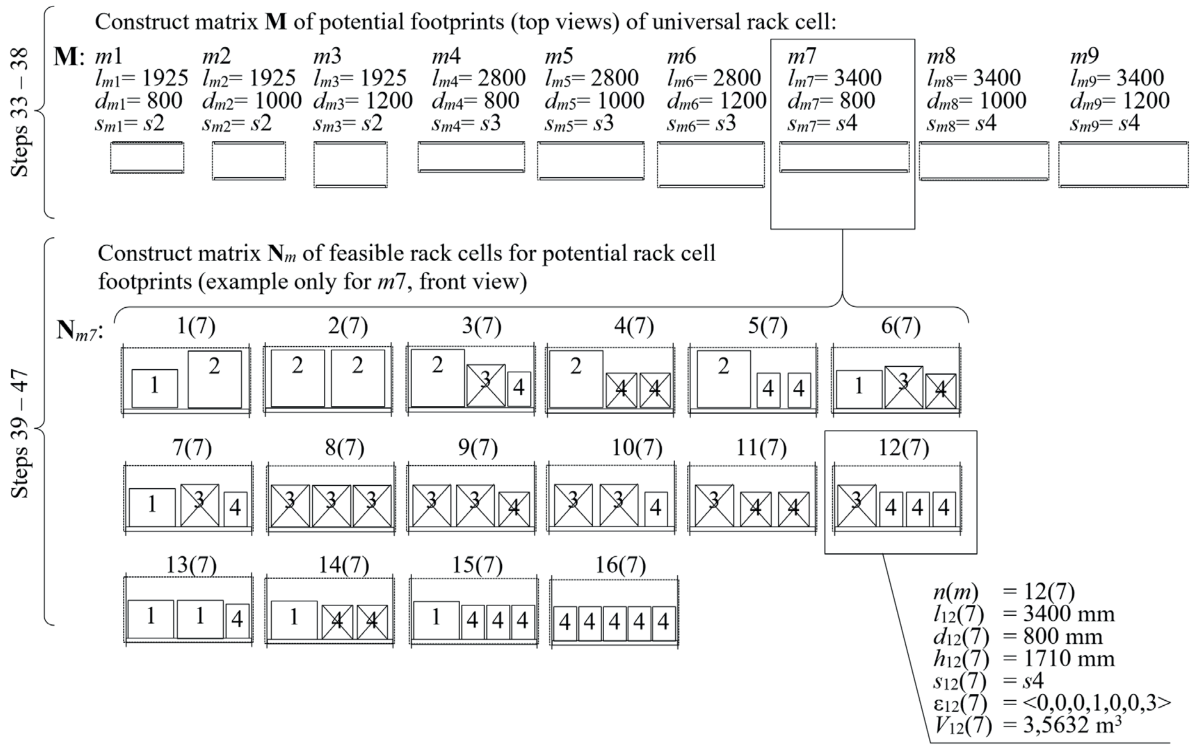

Fig. 7. Exemplary calculations – steps 33 to 47 of solving algorithm

39: For all m in **M** do
40: Create matrix $\mathbf{N}_m = NULL$ of vectors $n(m) = [l_n(m), d_n(m), h_n(m), s_n(m), \varepsilon_n(m)]$ //vector describing all possible item arrangements for m-footprint and height of rack cells resulting from these arrangements
where $\varepsilon_n(m) = [\alpha_f\colon \alpha_f \in \boldsymbol{C}^+ + \{0\}]$ //assign vector variants of item rotation to n-item arrangement on m-footprint.
41: While not all $\varepsilon_n(m)$ checked do //Until all possible item arrangements on a footprint are checked (brute force search)
42: find next feasible $\varepsilon_n(m)$
43: create new n_m
44: attribute $d_n(m) = d_m$; $l_n(m) = l_m$; $s_n(m) = s_m$; //attribute beam characteristics…
45: attribute $h_n(m) = \max\{h_i\} + g_s + b_3$ //rack cell height is from highest item…
46: If ($n(m)$ is complete) and ($n(m)$ is homogenous) then //see Comment 1
47: include $n(m)$ to $\mathbf{N}_m$

Comment 1:

- ***n_m* is complete** if the item arrangement $\varepsilon_{n,m}$ fully utilises the rack cell space, so no other item (rotated or not) can be placed in that rack cell,
- ***n_m* is homogenous** when the item arrangement $\varepsilon_{n,m}$ contains only items of the same type, but these are oriented in space in the same manner. If the rack cell can accommodate normally oriented and rotated i-type items at the same time, it can also accommodate these items oriented in the same manner, so this variant is redundant for other existing variants (Fig. 8).

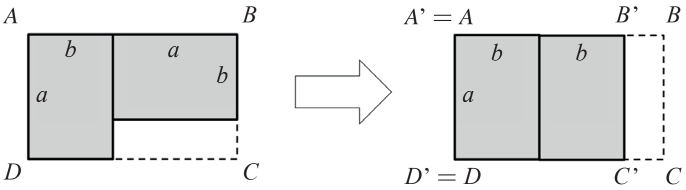

Fig. 8. Arranging items differently oriented in space

For each m in **M**, solve the system of integer linear inequalities (Fig. 9):
48: For all m in **M** do
49: compute integer linear inequalities to find space consumption $V(m)$ //see *Comment* 2
50: min $V(m) \to V^*$ //select the optimal footprint of a rack cell
51: Extract solution parameters for V^*
52: STOP

Comment 2: Solve the system of integer linear inequalities.

A. For each m-th footprint of the universal rack cell, solve the linear integer assignment problem:
   1. the system is composed of I inequalities (one for each item – Fig. 9),
   2. the left side of the inequality is a sum of N_m products. Each product is a product of the number of i-type items attributed to the n_m-th item arrangement, and the number βn_m of used arrangements of that type, where βn_m is an integer decision variable,
   3. the right side is a storage volume of i-type items λ_i, and must be equal to or smaller than the left side,
   4. the total cubic volume V_m of all selected n_m-th arrangements must be minimal.

B. Select the footprint that allows for storing all items in the minimal space, min V_m. Write all item arrangements with non-zero values βn_m for that footprint as a solution to the problem.

C. The branch-and-bound method is easy applicable and often provides global optimality for real cases.

4. Numerical study

In this section, the effects of the proposed model are investigated by means of a numerical example. For a given set of 30 item types (Table 1), set of eight types of rack beams (Table 2), and specified technical parameters of the storage area (Table 3), the URC must be determined to keep all items in the smallest

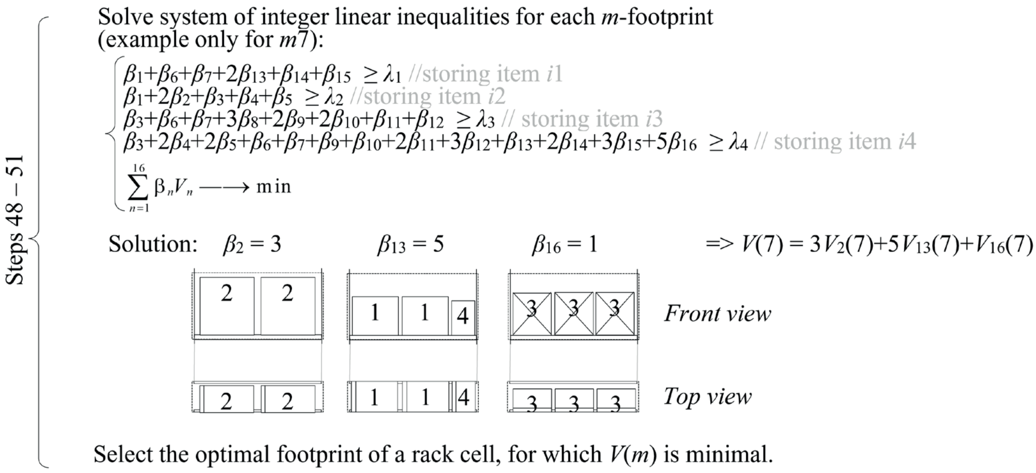

Fig. 9. Exemplary calculations – steps 48 to 51 of solving algorithm

Table 1
Set of items to be stored

Item	Volume λ_i	Dimensions (mm)			Weight c_i (kg)	Item	Volume λ_i	Dimensions (mm)			Weight c_i (kg)
		w_i	l_i	h_i				w_i	l_i	h_i	
$i1$	40	600	2600	1550	300	$i16$	260	950	1750	1500	700
$i2$	120	650	1750	1600	400	$i17$	50	1000	1550	1300	550
$i3$	400	700	1500	600	800	$i18$	220	1000	1700	1200	550
$i4$	295	700	1800	1100	750	$i19$	80	1000	2000	600	1100
$i5$	270	750	1850	1050	800	$i20$	230	1000	2600	1000	300
$i6$	70	600	900	1000	350	$i21$	60	1050	1500	2000	1100
$i7$	760	800	1200	1400	500	$i22$	40	1050	1650	950	1100
$i8$	300	800	1900	1500	750	$i23$	205	1100	1600	700	350
$i9$	270	800	2000	1500	550	$i24$	60	1150	1550	1200	800
$i10$	250	850	1850	1200	400	$i25$	60	1200	1500	950	750
$i11$	130	850	2000	950	1100	$i26$	70	1250	1450	1500	200
$i12$	210	900	900	1200	350	$i27$	160	1300	1200	1200	400
$i13$	210	900	1800	950	300	$i28$	170	1300	1350	950	300
$i14$	280	900	2000	900	550	$i29$	90	1400	1000	1500	700
$i15$	230	950	1200	1500	700	$i30$	235	1600	1000	950	800

Table 2
Rack beam types

Type of rack beam s	$s1$	$s2$	$s3$	$s4$	$s5$	$s6$	$s7$	$s8$
Length l_s (mm)	1825	2225	2625	2700	2700	3300	3600	4200
Height – thickness g_s (mm)	60	80	90	110	140	110	140	165
Loading capacity c_s (per pair/kg)	1700	2000	2000	2000	3600	2000	2000	2900

Table 3
Technical parameters of storage area

Maximum depth of rack cells $d^{\max}$	1450 mm
Width (thickness) of rack pillar b_1	150 mm
Vertical safety gap between items or unit and rack pillar b_2	50 mm
Horizontal safety gap between item and rack beam b_3	100 mm

cubature. Provided rack beams and technical parameters of the storage area are typical for most situations.

The algorithm proposed in Section 3.2 was implemented to perform calculations on real examples, check the feasibility conditions and visualise the results. A summary of the calculation results for the above data is presented in Table 4 and discussed in the following section – the optimal solution is bolded. Several feasible rack cells can host additional items beyond those required, as additional profit.

Beams s = 1, 2, 3 do not secure proper storage of all item types when the maximal cell depth is set to 1450 mm. The first type of beam to be used is s = 4. The results demonstrate that

Table 4
Brief results sheet

Type of beam s	Rack cell dimensions		Rack cell height, h = 2240 mm		
	l (mm)	d (mm)	Number of cells to create storage area	Storage area cubic volume (m^3)	Additional items that fit in storage area
1	–	–	No feasible solution		
2	–	–	No feasible solution		
3	–	–	No feasible solution		

Type of beam *s*	Rack cell dimensions		Rack cell height, h = 2240 mm		
	l (mm)	*d* (mm)	Number of cells to create storage area	Storage area cubic volume (m^3)	Additional items that fit in storage area
4	2850	1300	4395	35986.5	1935 of *i*6
		1350	4395	37370.6	1935 of *i*6
		1400	4305	37961.1	1755 of *i*6
		1450	4265	38951.5	1675 of *i*6
5	2850	1300	4395	36475.0	1935 of *i*6
		1350	4395	37877.9	1935 of *i*6
		1400	4305	38476.4	1755 of *i*6
		1450	4265	39480.3	1675 of *i*6
6	3450	1300	3510	34790.6	1075 of *i*6
		1350	3510	36128.7	1075 of *i*6
		1400	3510	37466.8	1075 of *i*6
		1450	3510	38804.9	1145 of *i*6
7	**3750**	**1300**	**2749**	**30019.1**	**64 of *i*6; 2 of *i*12**
		1350	2749	31173.7	73 of *i*6; 1 of *i*15; 1 of *i*17
		1400	2749	32328.2	74 of *i*6; 2 of *i*12
		1450	2749	33482.8	73 of *i*6; 1 of *i*12; 1 of *i*18
8	4350	1300	2357	30189.8	1 of *i*6; 1 of *i*15; 1 of *i*21
		1350	2357	31351.0	1 of *i*6; 1 of *i*15; 1 of *i*23
		1400	2357	32512.1	1 of *i*12; 1 of *i*16
		1450	2357	33673.3	1 of *i*7; 1 of *i*16

the given structure of items is not conducive to short beams (Fig. 10). In most cases, short beams result in space wastage because only one or two items can be placed in a single cell. Short beams also increase the number of rack cells, so additional space is wasted for rack construction elements (Fig. 10). A large number of short beam cells may be partially balanced by less space consumed per cell, but the final settlement is adverse for this option.

A diversified structure of items (as in the example) requires longer beams. Numerous variations of item arrangements provide a superior fit to the cell dimensions. The optimal result $(30019.1\ m^3)$ is approximately 23.9 % better than the worst result $(39480.3\ m^3)$ for a shorter beam. This is owing to the general rule that a greater difference between object sizes results in a smaller relative loss of space when filling a larger object with smaller objects.

A fixed cell height results in an inevitable waste of space above stored items. The highest item determines the cell height, so placing low items on a long beam is not optimal. This is presumably the reason why beam $s = 7$ (not the longest) is selected. The shallowest cell (1300 mm) is proven to be rational. This can be explained analogously to the cell height. The deepest stored item influences the cell depth, but when it can be stored rotated, its effect is lowered (for example, $i = 26$ or $i = 29$). Of course, this does not apply to items that must be stored rotated, according to technical constraints (such as $i = 1$ or $i = 9$).

Most types of universal rack cells allow for storing additional items. This is the result of a restriction stating that item arrangements that fail to fill the cell space fully are rejected. This is additional profit (for example, 1935 of items of 6-th type for beam of 4-th type), but it is difficult to estimate without detailed characteristics of the stored materials (Fig. 11).

The rational solution (bolded in Table 4) is detailed in Table 5. A total of 2749 cells with dimensions of 3750×1300×2240 mm and a cubic capacity of 10.92 m^3, based on the 7-th type of rack beam, allow for storing 5825 different items, as described in Table 1, in the minimal possible space. The full solution for

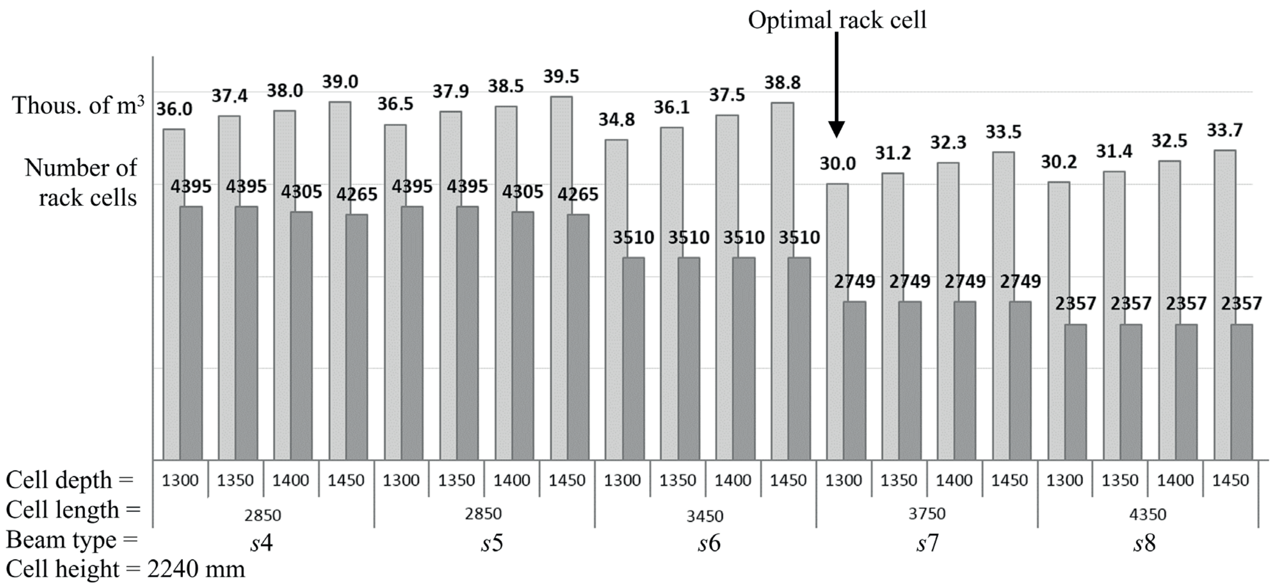

Fig. 10. Total space consumption and number of rack cells for different feasible solutions

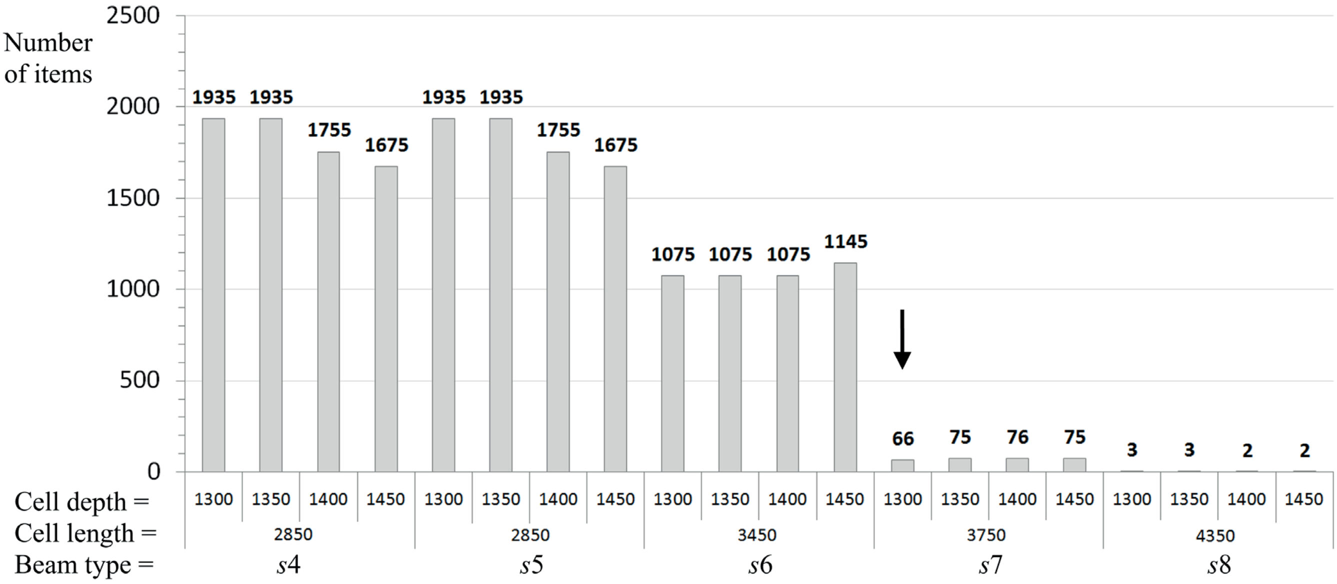

Fig. 11. Number of additional items fitting storage area in different feasible solutions

Table 5
List of admissible item arrangements in rack cell with dimensions 3750×1300×2240 mm (rational)

Total cubic volume of items in n-th variant of arrangement (m^3)	Total weight of items in rack cell (kg)	Variants of item arrangements in rack cell (* = item is stored rotated)	Number of used rack cells storing n-th variant of item arrangements β_n
3.76	800	1 of *i*1*; 1 of *i*7	40
3.86	950	1 of *i*2*; 1 of *i*18*	20
3.47	1500	1 of *i*2*; 1 of *i*22*	40
3.53	1150	1 of *i*2*; 1 of *i*25*	60
2.91	1550	1 of *i*3*; 1 of *i*8*	160
2.52	1200	1 of *i*3*; 1 of *i*10*	230
3.12	1500	1 of *i*3*; 1 of *i*16*	10
2.91	1550	1 of *i*4*; 1 of *i*30	235
2.69	1150	1 of *i*5*; 1 of *i*23*	201
3.94	800	1 of *i*7; 1 of *i*20*	230
5.43	1850	1 of *i*8*; 1 of *i*21*	60
4.42	1550	1 of *i*8*; 1 of *i*24*	60
4.27	950	1 of *i*9*; 1 of *i*27*	100
4.07	850	1 of *i*9*; 1 of *i*28*	170
3.49	1500	1 of *i*11*; 1 of *i*27*	40
3.72	1800	1 of *i*11*; 1 of *i*29	90
2.17	1450	1 of *i*12; 1 of *i*19*	80
3.33	1250	1 of *i*14*; 1 of *i*15*	210
4.34	750	1 of *i*14*; 1 of *i*26*	70
4.51	1250	1 of *i*16*; 1 of *i*17*	50
4.53	1250	1 of *i*16*; 1 of *i*18*	200
2.90	1450	1 of *i*4*; 1 of *i*6; 1 of *i*12	25
4.07	1750	1 of *i*4*; 2 of *i*7	35
2.97	1500	1 of *i*5*; 1 of *i*6; 1 of *i*12	69
3.79	1450	1 of *i*6; 1 of *i*8*; 1 of *i*12	20
4.14	1450	1 of *i*6; 1 of *i*10*; 1 of *i*15	20
4.23	1300	2 of *i*7; 1 of *i*13*	210
3.18	1050	2 of *i*12; 1 of *i*23*	4
4.72	1150	1 of *i*12; 2 of *i*27*	10
			Total: 2749 cells

that cell consists of 552 β_n values (only non-zero values are listed).

It should be noted that, in most cases in Europe, racking systems in wholesale are configured to handle ISO1 1200×800×144 mm pallets. The usual rack cell for such a pallet uses a 2700 mm beam and is approximately 1300 mm deep. As can be observed in Table 4, the storage area constructed on this basis consumes 36475 m^3, while the storage area constructed based on the rational solution consumes 30019.1 m^3. The rational solution is therefore approximately 21% better than the common feasible solution applied in wholesale.

The proposed algorithm is time-effective and provides an optimal solution for large-scale cases. The first stage of the method is based on brute force search to identify all possible rack cells (bins). These results are used in the second stage to allocate items in the bins and determine the number of particular bins. Optimality is achieved when system of integer linear inequalities in the second stage is solved optimally. The proposed branch-and-bound method provides global optimality in approximately 98% of trials. The remainder requires reconstruction of the inequalities set to determine global optimality in the following attempts. This involves solving the task by the simplex method, and then generating *Gomory's* cuts [14] to provide global optimality in every attempt.

5. Conclusions

The most important result of this paper is new and effective method for determining rack cell dimensions for heterogeneous storage. The resulting rack cell is a base for the modular design and operation of a storage area, similar to homogenous storage. The proposed approach allows for the use of known methods and techniques to optimise the storage area for heterogeneous items, by means of minimising space, increasing flexibility, and reducing costs.

The optimal space utilisation level can be achieved when all stored items have storage space fitted exactly to the dimensions, but this requires uniform items and a non-changeable material flow volume. In general, the structure of material stock changes more quickly than storage areas can be reconfigured, so it is rational to build a storage area that is as universal and modular as possible.

The proposed model provides a practical tool that can be used during the stage of designing warehouses or retail sale points. It is a tool for gaining additional space profits that create new possibilities under conditions of increasing competition. Inevitable and on-going displacement of the assortment, and its dimensions and weights, can lead to a decrease in space utilisation in racking systems. Therefore, it becomes necessary to reconfigure the rack cells. The simplest means of fitting existing rack system to new requirements is height adjustment of particular rack cells, which should be a next step in further research.

The method can also be used in other areas where it is necessary to arrange a set of items in a space, e.g., locating equipment in a limited cargo space. It may be then an instance of a wide class of backpack problems with new type of constrain for bin size.

Acknowledgements. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] M. Kłodawski, K. Lewczuk, I. Jacyna-Gołda, and J. Żak, “Decision making strategies for warehouse operations”, *Arch. Transp.* 41(1), 43–53 (2017).

[2] I. Jacyna-Gołda, M. Izdebski, E. Szczepański, and P. Gołda, “The assessment of supply chain effectiveness”, *Arch. Transp.* 45(1), 43–52 (2018).

[3] M. Jacyna , M.Wasiak, and A. Bobiński, “SIMMAG3D as a tool for designing of storage facilities in 3D”, *Arch. Transp.* 42(2), 25–38 (2017).

[4] K.R. Gue and R.D. Meller, “Aisle configurations for unit-load warehouses”, *IIE Trans.* 41(3), 171–182 (2009).

[5] S. Labant, M. Bindzárová Gergel’ová, Š. Rákay, E. Weiss, and J. Zuzik, “Track planarity and verticality of the warehouse racks for the quality assessment of further operation”, *Geodesy Cartogr.* 68(2), 305–319 (2019).

[6] G. Dukic and T. Opetuk, “Warehouse layouts”, in *Warehousing in the Global Supply Chain. Advanced Models*, *Tools and Applications for Storage Systems*. (Ed.) Manzini, R., pp. 55–69, Springer-Verlag, London, 2012.

[7] G. Kovács, “Layout design for efficiency improvement and cost reduction”, *Bull. Pol. Ac.: Tech.* 67(3), 547–555 (2019).

[8] T. Lerher and M. Sraml, “Designing unit load automated storage and retrieval systems”, in *Warehousing in the Global Supply Chain. Advanced Models*, *Tools and Applications for Storage Systems*. (Ed.) Manzini, R., pp. 211–231 Springer-Verlag, London, 2012.

[9] H.L. Lee, M.H. Lee, and L.S. Hur, “Optimal design of rack structure with modular cell in AS/RS”, *Int. J. Prod. Econ.* 98(2), 172–178 (2005).

[10] A. Ratkiewicz, “A combined bi-level approach for the spatial design of rack storage area”, *J. Oper. Res. Soc.* 64(8), 1157–1168 (2013).

[11] H. Dyckhoff, “Cutting and packing in production and distribution: a typology and bibliography”, Springer-Verlag, Berlin, 1992.

[12] G. Wäscher, H. Haußner, and H. Schumann, “An improved typology of cutting and packing problems”, *Eur. J. Oper. Res.* 183(3), 1109–1130 (2007).

[13] E. Silva, J.F. Oliveira, and G. Wäscher, “2DCPackGen: A problem generator for two-dimensional rectangular cutting and packing problems”, *Eur. J. Oper. Res.* 237(3), 846–856 (2014).

[14] S. Martello, “Packing problems in one and more dimensions”, in *Winter School on Network Optimization*, *7th edition*, 2018, Estoril, Portugal. [Online]. Available: http://www.or.deis.unibo.it/staff_pages/martello/Slides_Estoril_Martello.pdf (accessed: May 01, 2020]

[15] G. Scheithauer, “Introduction to cutting and packing optimization”, *International Series in Operations Research and Management Science*, Springer-Verlag, Berlin, 2018.