In the simulation an ideal sixport correlator is used. Hence by implementing the broadband phase shifting network instead of the $\lambda/4$ TL the carrier leakage can be suppressed over a much wider bandwidth. For an EVM of less than 10% requires $E_f \leq -14$ dB. An EVM of less than 10% is achived with the broadband phase shifting network over a relative bandwidth of about 60%, to compare with the relative bandwidth of about 12% for the $\lambda/4$ TL phase shifting network.

Conclusion

The performance of the carrier leakage suppression and the modulation performance in terms of EVM were further investigated as a function of the phase shifting network. Both carrier leakage suppression and the EVM performance can be described by the same error function. The error function is directly related to the amplitude and phase behavior of the phase shifting network, i.e., it is related to the S-parameters of the phase shifting network. For wideband performance, a loaded TL was proposed as one possible solution to implement the phase shifting network. It was designed and optimized with help of the derived error function.

References

1. Semezhev N., Multiport Modulator/ N. Semezhev, J. Östh, A.A. Sytnik // Problems of control, information processing and transmission (CIPT-2017) Proceed. of the Internat. Scientific Conf., pp. 581-584, 2017

2. Osth, J. Carrier leakage suppression and evm dependence on phase shifting network in six-port modulator/J. Osth, O.M. Karlsson, A. Serban, S. Gong // Proc. Int. Conf. Microwave and Millimeter Wave Technology (ICMMT), vol 4., 2012.

3. Gu, Q. RF System Design of Transceivers for Wireless Communications./ Q. Gu // Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

4. Zheng, S. Y. Broadband phase shifter using loaded transmission line/ S. Y. Zheng, W. S. Chan, K. F. Man // IEEE Microwave and Wireless Components Letters, vol. 20, no. 9, pp. 498–500, 2010.

M. Verkhoturov, G. Verkhoturova, R. Yagudin, K. Danilov

NO-FIT POLYHEDRON FOR IRREGULAR PACKING OF NON-CONVEX OBJECTS

(Ufa State Aviation Technical University)

1. Introduction

The emergence of additive technologies and rapid prototyping techniques revolutionized the high-tech industries, for instance aviation and aerospace industry, nuclear industry, medical and instrumentation. They are characterized as small-scale or piece production. Using new methods for the synthesis of forms and synthesis models by layering synthesis technology allowed to drastically reduce the time to create new products. Since a number of independent parts can be manufactured simultaneously, the implementation of such technologies leads to the necessity of solving the problem of the irregular 3D objects placement optimization, which is desirable from the standpoint of saving time, energy and other resources.

2. Statement of a problem

Suppose we have a set of 3D geometric objects (GO) $T = \{T_1, T_2, ..., T_n\}$: $T_i \subset \mathbf{R}^3, i = \overline{1,n}$, each in its own coordinates.

Layout area $Q \subset \mathbf{R}^3$ is a rectangular parallelepiped with variable height H, fixed length L and a width W.

Let $T_i(\bar{u}_i)$ is a geometric object T_i offset by vector $\bar{u}_i(x_i, y_i, z_i)$.

Resulting positioning schema must fulfill the following conditions:

- Mutual non intersection:

$T_i(\bar{u}_i) \cap T_j(\bar{u}_j) = \varnothing, \forall i = \overline{1,n}, \forall j = \overline{1,n}, i \neq j$

- Being inside container

$T_i(\bar{u}_i) \cap Q = T_i(\bar{u}_i), \forall i = \overline{1,n}$

Equations (1) and (2) restrict possible placement parameters $U = (\bar{u}_1, \bar{u}_2, ..., \bar{u}_n) \in R^{3n}$ for objects set T inside area Q.

Let H = Z(Q(U)) to be minimal height to place all objects of $T = \{T_1, T_2, ..., T_n\}$ with offset vectors $U = \{\bar{u}_1, \bar{u}_2, ..., \bar{u}_n\}$.

Problem is to find a set of offset vectors U that minimize Z(T(U))->min, while restrictions (1) and (2) remains met.

In above terms, this problem is complex optimization of geometric modeling in high-dimensional space with nonconvex and disconnected space of possible solutions. It belongs to NP complexity class. In addition to optimization, it has also geometric aspect to obey restrictions of mutual non-intersection and placement inside given layout space, Stoyan et al. (2009).

3. Problem approaches

Popular methods for solving 2D and 3D tasks of complex shaped geometric objects irregular placement are those of rational (permissible) pilings close to optimal. Usually they operate with single object at every single step of decision (object by object placement principle).

Solution process consists of the following procedures, named "encoding", "decoding" and "evaluating", Lutters (2012):

1. Optimization - ordering sequence of objects:
   - Generation of sequence of objects to place;
   - Reordering of objects;
2. Geometric procedure applied to objects according to their position in sequence:
   - Appropriate object representation (polygonal, voxel etc.);
   - Object motion modeling;
   - Choosing object position according to some criteria
   - Object placement into area with possible area growth

These procedures are often thus combined:

1. Generating object sequence (ordered list)
2. Sequence loop

2.1. Object motion modeling
2.2. Choice of object position according to some criteria
2.3. Adding object to area (with possible area growth)
3. Calculating goal function

The loop is terminated after predefined iterations, time or when goal function reaches its limit.

A large variety of heuristics used for solving irregular placement problems at optimization phase exist. In most cases two methods classes are used. The first one is metaheuristics like "simulated annealing" (SA), "genetic algorithm" (GA), "tabu search" (TS), "ant colonies" (AC) with their modifications. The second one is heuristic methods crafted specifically for these problems.

Geometric procedures can be implemented in three ways:

1. Simulating object motion with mutual non-intersection (inside layout area)
2. Arbitrary motion (shifts and rotations), where object can overlap each other and layout area
3. Positioning objects into arbitrary area

These methods differs in:

- Path of object movement
- Complexity of rotation modeling
- Whether object intersections are allowed during solution phases

The one of the most wide used geometric methods is based upon modeling object movements inside layout area with restriction of their mutual non-intersection. It uses the concept of No-Fit-Polyhedron (NFP), Egeblad et al. (2007).

No-Fit-Polyhedron G_{12} or $G(T_1(0), T_2(u_2))$ for moving object $T_2(u_2)$ around fixed object T_1 is the set of T_2 positions where it is tightly fit to T_1.

3.1 NFP usage scenarios for object placing considering already placed objects and layout area

Several approaches for using NFP are known, Verkhoturov (2012):

1. **Preliminary**. NFP for all object pairs and layout area are calculated beforehand. After object positioning, all NFP involved also shift according its new position.

2. **Integral**. For every object its NFP is calculated, as if already positioned objects were parts of layout area.

The main disadvantage of the first approach is that it assumes a lot of NFP calculation which will never be used.

The second approach often leads to unconnected layout area, that makes difficult to find available positions to place next object.

“Dynamic” NFP scheme was developed to overcome these drawbacks. It allows avoiding excessive NFP calculations.

3. **Dynamic**. NFP for object to place is calculated for layout area and subset placed object. Then every NFP is restricted using aforementioned package conditions.

NFP algorithm with dynamic scheme

Suppose first (m-1) objects $\{T_1, T_2, \ldots, T_n\}$ are already placed, having m-1<n. The next step is to position T_m object as follows:

1. For T_m object its NFPs are calculated for objects of ordered list $K=\{K_0, \ldots, K_{m-1}\}$. $K_0 = Q$, a $\{K_1, \ldots, K_{m-1}\}$ is reordered list of placed objects $\{T_1, \ldots, T_{m-1}\}$, sorted by ascending position height (Fig. 1a).

2. After calculating every NFP $G_i(K_i, T_m)$, its points $\{u_i\}$ are filtered (Fig. 1b) using condition: $u_i \notin \operatorname{int} G_j(K_j, T_m), \forall j = \overline{0, m-1}, j \neq i$

Condition check $u_i \notin \operatorname{int} G_j(K_j, T_m)$ can be safely skipped for some K_j when surrounding cuboids of K_j and T_m have no intersection (Fig. 1c).

3. If some u_i found available (Fig. 1d), NFP calculation can be skipped for $\{K_j\}$, having:

$$\min Z(K_j) > \max Z(T_m(u_i))$$

During calculations, when "small" objects are positioned after "big" ones according to sorted list order, they make placement more dense by arranging "in the bottom". The proposed approach thus allows make last steps faster by eliminating most NFP computations.

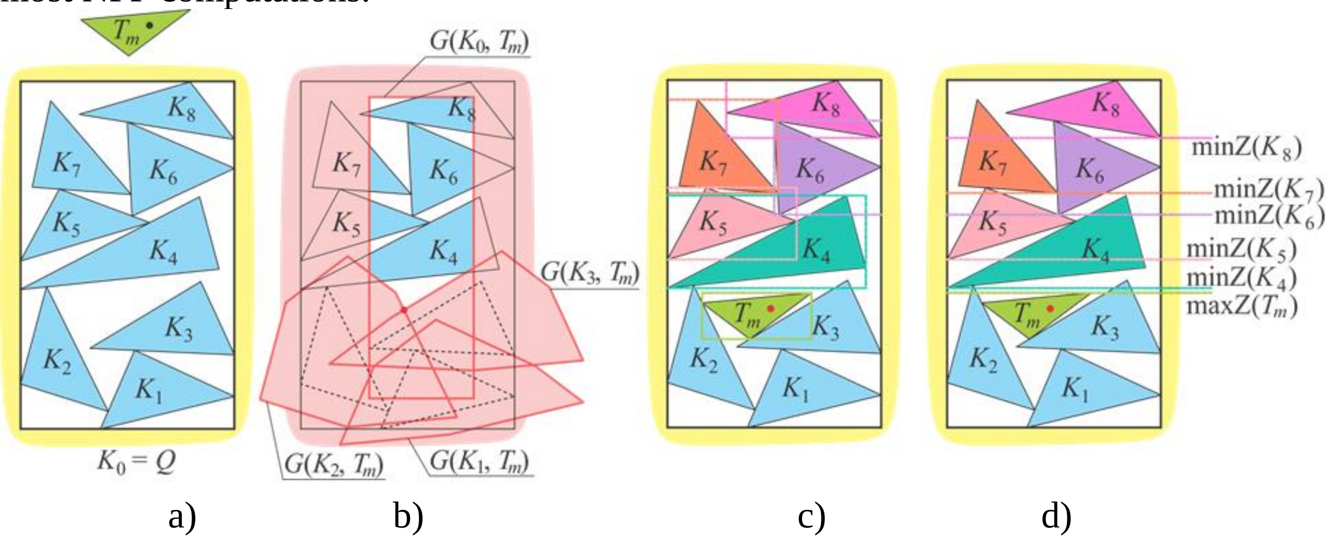

Fig. 1 Dynamic NFP scheme

3.2 NFP GENERATION

The analysis of NFP application methods (Fig. 2) leads to the following conclusion: those methods consistently changed from using floating point operations to integer arithmetic and further on. Simplification of basic operations, taking into account need of their reliability increase, is possible with transition to logical actions. Feature of this representation is that only logical operations over 0 and 1 are necessary for calculation of geometrical objects crossing.

Algorithm of NFP outer hull determination (object-base representation)

This algorithm is developed based on the algorithm "Pseudo Faces", Chernomorets (1993). The developed algorithm is not built of NFP inner since it is not really necessary for three-dimensional packing.

Given two objects T_1 и T_2 (Fig. 3). It's necessary to determine outer component $\overline{G}_{12}$ of NFP G_{12}: $\overline{G}_{12} \subset G_{12}$

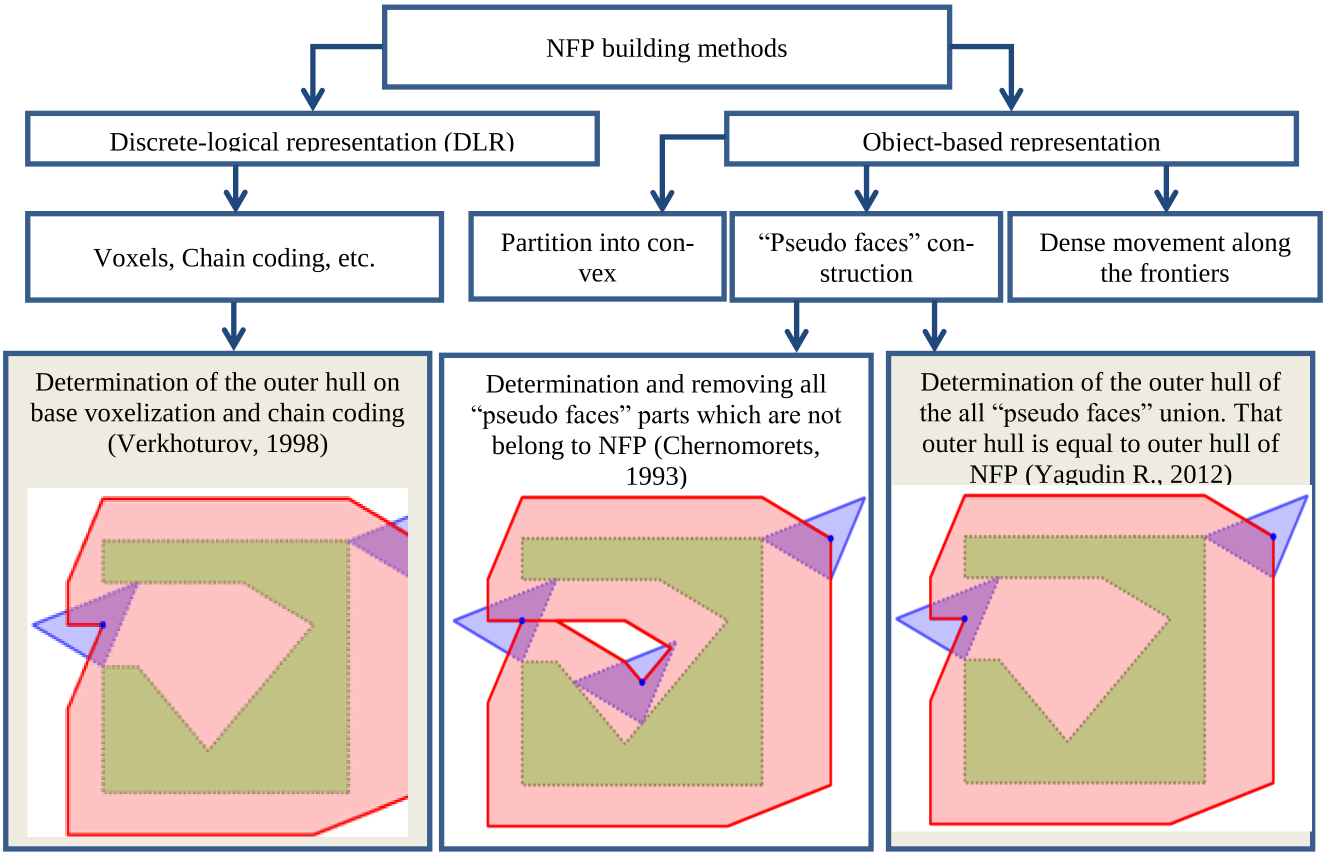

Fig. 2 NFP building methods using object-base and voxel-base representation (2D case)

1. Firstly need to consider four contact variants of polyhedrons (variants “Vertex” – “Vertex” and “Vertex” – “Edge” are their subset):
- “Face” – “Face” (Fig. 3a);
- “Face” – “Edge” (Fig. 3b);
- “Face” – “Vertex” (Fig. 3c);
- “Edge” – “Edge” (Fig. 3d)

and then construct a set of “pseudo faces” $\{s_i\} = S$ of NFP G_{12} $G_{12} \subset S, S \setminus G_{12} \subset \operatorname{int} G_{12}$ (Fig. 3e)

2. Second step is to determine which of “pseudo faces” belongs to the outer hull of object S.

3. Recursively bypass object S from the outside. This bypass starts from definitely outer “pseudo faces” and grabs other faces fully or practically.

4. The union bypassed parts of the object S is equal to desired outer hull $\overline{G}_{12}$ of NFP G_{12}(Fig.3f).

Algorithm of NFP determination (voxel-base representation)

The basic idea of this approach is "direct" simulation of a solid motion of objects in a computer memory. That is, main operations of NFP construction (shift, choice of motion direction, calculation of intersection etc.) are performed using discrete-logical structure of computer memory. Three-dimensional NFP can be built using discrete-logical representation in many ways depending on:

- Object boundaries connectivity (6, 18 or 26-fold for 3D)
- Contact of object boundaries with packing area (“tight” or “loose”)
- Choice of object shift direction

3D objects surfaces are represented as set of the partial vectors focused in 6, 18 or 26 directions depending on the chosen principle of coding, Verkhoturov et al. (2000).

This is due to the fact that in computer memory representation any non-edge element has six, eighteen or twenty six adjacent element depending on used diagonal directions (Fig. 5).

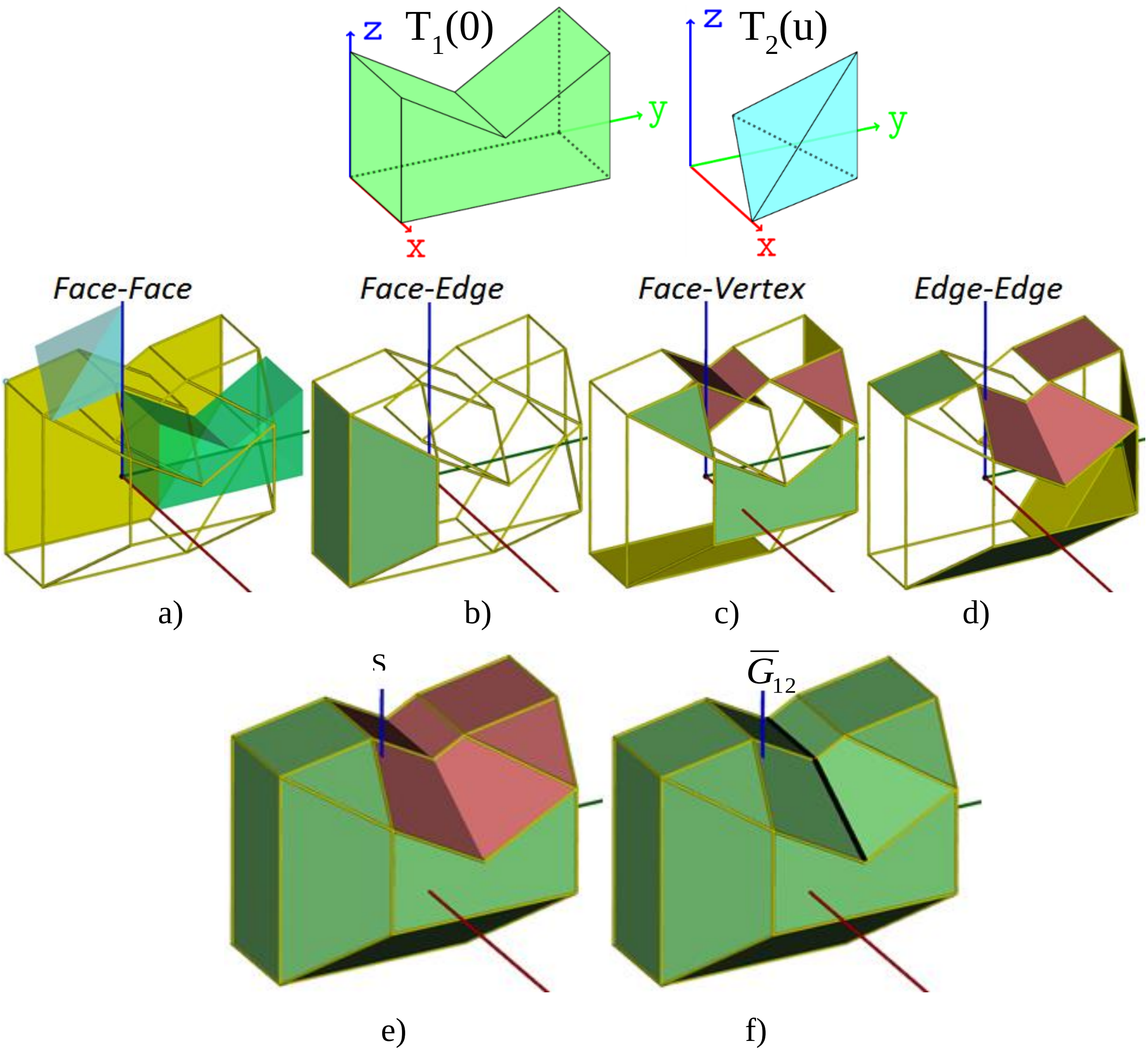

Fig. 3. NFP outer hull determination

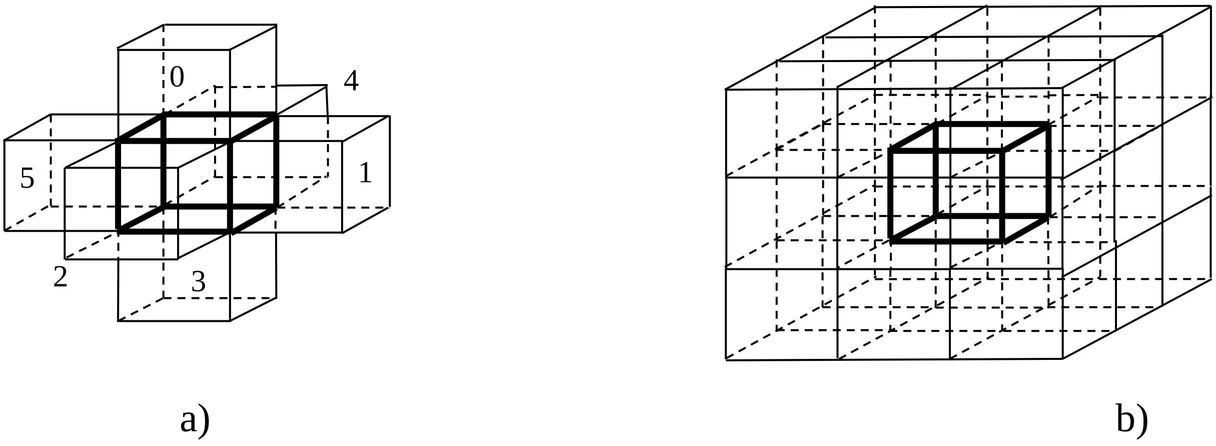

Fig. 4 Voxel-base representation
a) 6-fold b) 26-fold

Six-fold coding is the easiest representation of 3D objects surface and most reliable for NFP construction, for it makes impossible “diagonal penetration” to occur, Verkhoturov (1998).

Eighteen- and twenty six-fold coding allow shorter vectors list to represent objects.

Voxel-base representation allows NFP construction with different accuracy.

Choice of object motion direction during NFP construction

Unlike 2D case, motion modelling for 3D objects is far more difficult task, for there is no clear evidence where and how object should be moved to get around all the points of the area. To solve this problem, we proposed and developed an approach based on "Fill solid areas with seed voxel" and "Depth-first search" algorithms (Fig. 5).

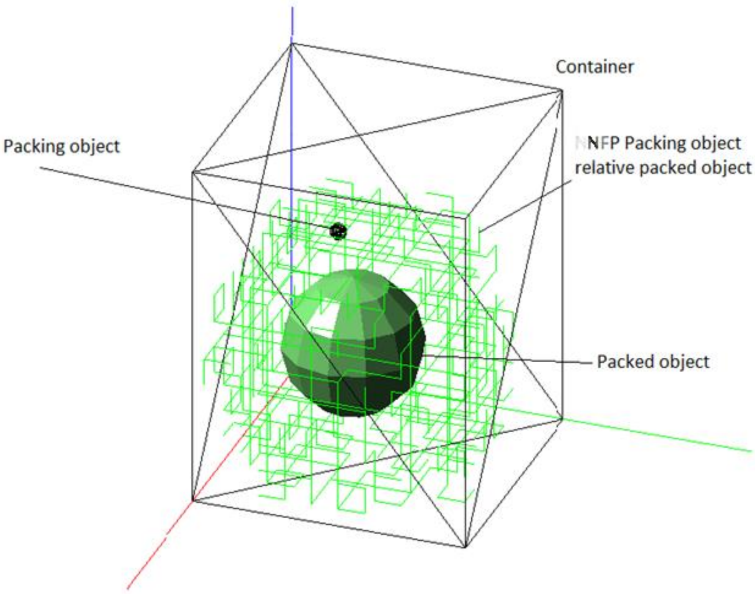

Fig. 5 NFP construction for packing object

4. Computing experiment results

For quality check of the methods and algorithms developed during this study the computing experiment was made with sample data available in public and practical cases. The results were also compared with other methods.

For an assessment of effectiveness the data from Y.Stoyan (2004) and J. Egeblad (2009) articles were used.

Samples 1-3: Sets from ten polyhedra: 20, 30 and 40 (two, three and four of each type). Packing area base is 30x35. Comparison was made by packing density (%). Results are at Fig. 6.

Algorism	Packing density, %		
	20 obj.	30 obj.	40 obj.
First local minimum (Y. Stoyan)	17.71	19.7	19.03
Decremental neighborhood search (Y. Stoyan)	24.2	23.71	24.5
Random search (Y. Stoyan)	21.75	23.71	23.37
First Fit (FF) (object-base)	17.14	19.42	23.03
First Fit (FF) + Local Search (object-base)	22.8	22.91	25.61
«GRASP» (object-base)	19.32	18.54	17.89
«GRASP with Local Search»	24.47	21.78	20.53
Simple heuristic (voxel-base)	24.01	24.09	25.55

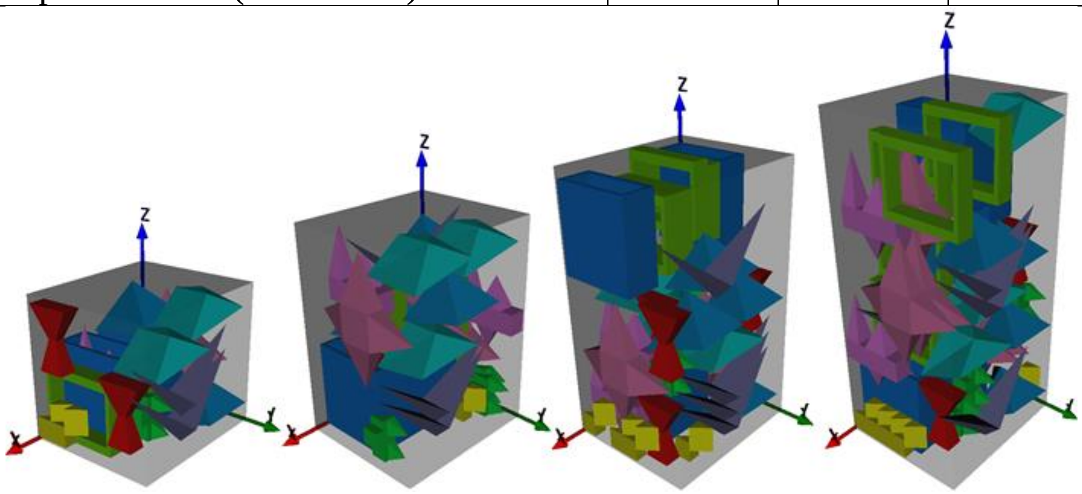

Fig. 6 Algorithms comparisons for samples 1-3

Samples 4-8: Sets from fifteen polyhedral: 15, 30, 45, 60 and 75 (one, two, three, four and five of each type). Packing area base are: 15x15, 17x17, 22x22, 24x24 and 26x26 accordingly. Comparison was made by packing density (%). Results are at Fig. 7.

(Notice: "SS" - "Smart space" is packing module of Magics software developed by Materialise Company).

The figure shows that in most cases the best packing density is achieved using object-base representation ("The first fit with ordering + LS" and "GRASP + LS" algorithms). The density of objects packing obtained by the voxel-base representation is somewhat lower because simplest implementation of the optimization procedure has been used, however at particular parameters of accuracy it allows to pack objects faster.

Results obtained from computational experiment lead to the following conclusions.

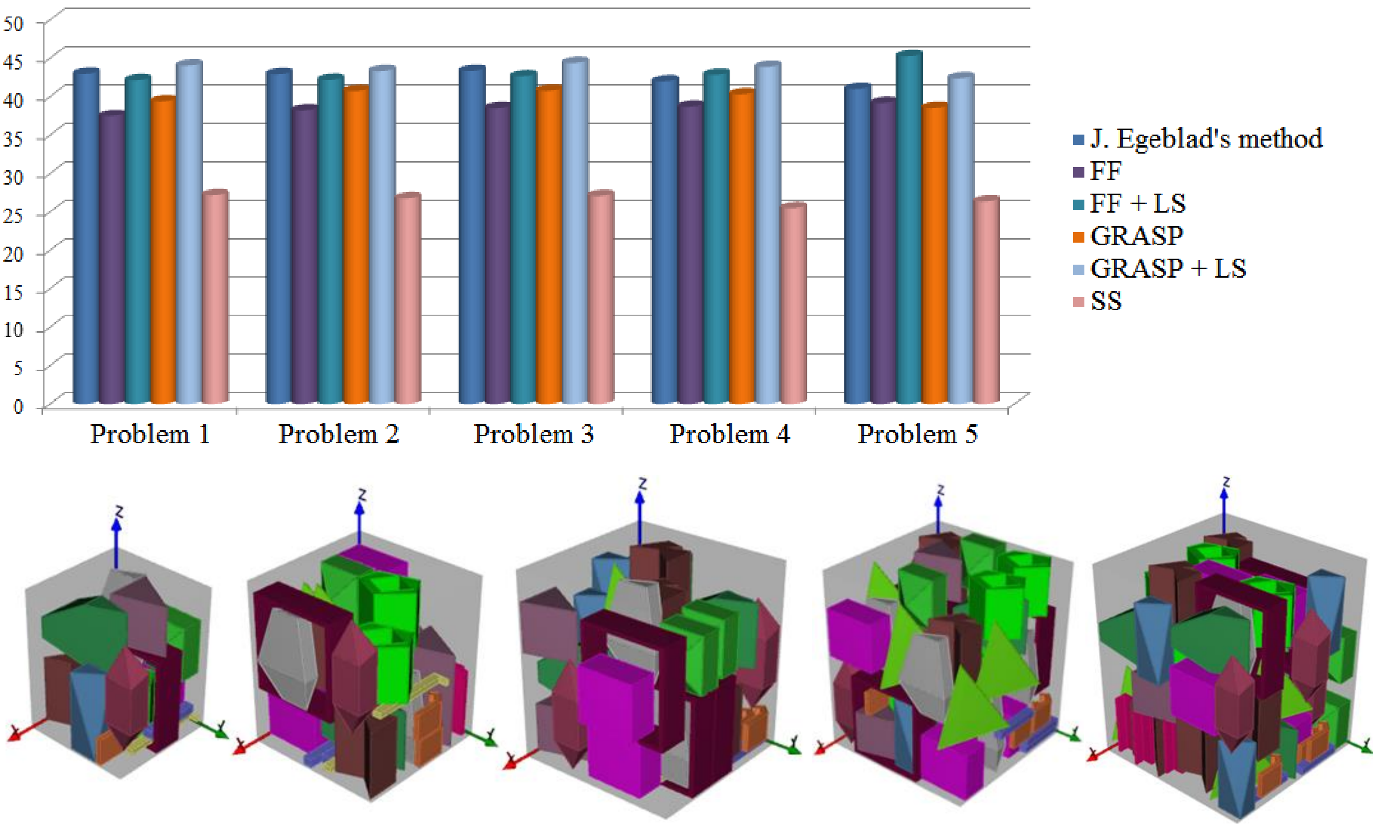

Fig. 7 Algorithms comparisons for samples 4-8

Main advantages of voxel-base representation are:

- Solution correctness (in this view): small changes in the source data do not entail a change in the results
- Speed and reliability of realization of basic logical operations

Ability to control resulting accuracy: depending on the chosen admission of approximation (a step of a discrete-logical grid) it is possible to receive rough (for initial solution steps) and precise results (for a final solution). When the faces number grows to thousands, floating point calculations reliability sharply falls, whereas DLR operation (voxel-base representation) is not affected in any way.

5. Conclusion

The paper considers approaches to solving the problem of packing non-convex polyhedra into a parallelepiped container based on the NFP construction using object-base and voxel-base representations, allowing a variety of results in term of time spent and accuracy. Package density at increase in objects accuracy with the use voxel-base representation approaches shared results. In addition, these studies have demonstrated that package time at the use is de facto independent on polygonal approximation accuracy, though the latter has a significant impact on resulting quality.

References

1. Stoyan,Yu., Chugay, A. Packing cylinders and rectangular parallelepipeds with distances between them into a given region. European Journal of Operational Research 197 (2009) 446-455.
2. Jens Egeblad, Benny K. Nielsen, Allan Odgaard. Fast neighborhood search for two and three-dimensional nesting problems. European Journal of Operational Research 183 (2007) 1249–1266.
3. Lutters, E., D. ten Dam, T.Faneker. 3D nesting of complex shapes. (2012) 3D Nesting of Complex Shapes. Procedia CIRP, 3 . 26 - 31. ISSN 2212-8271.
4. Stoyan Yu., Gil N., Scheithauer G., Pankratov A. Packing non-convex polytopes into a parallelepiped. TU Dresden, 2004.-32c.- (Preprint MATH-NM-06-2004)
5. Chernomorets A. Mathematical model and solution method of polyhedra packing problem, Kharkov, IPMash, 1993.-16p.
6. Verkhoturov, M.A., Verkhoturova, G.N. Process modeling of 3D packing problem on base discreet logical representation. – Deciding under Conditions of Uncertainty, SATUU, Ufa, 285-293, 1998.
7. Gil N, Chernomorets A. Built method of surface 0-level F-function. - Kharkov, IPMash, 1999.-29p. (Preprint ASU-IPM-329-1999)
8. Verkhoturov, M.A., Verkhoturova, G.N., Yagudin, R.R. Geometric optimizing modeling in the systems of 3-D objects placing. Workshop on computer science and information technologies CSIT’2012, Ufa – Hamburg – Norwegian Fjords, 2012, pp. 176 – 179.
9. Yagudin, R.R. (2012) Challenge to optimize packing polyhedral in parallelepiped region by constructing vector hodograph vector-function dense placement. Scientific Technical statements. St. Petersburg State Polytechnical University. System analysis and management. vol.5(157), pp.58-63. SP-bSPU, St.Petersburg.
10. Verkhoturov, M.A., Verkhoturova, G.N., Yagudin R.R. (2013). Dynamic usage of no-fit polyhedron for solving the non-convex polyhedrons dense packing problem. A series of mathematical and computer modeling. vol.5(79), pp.15-23. OGTU, Orel.
11. Egeblad J., Benny K., Marcus B. Translational packing of arbitrary polytopes. Elsevier. Computational geometry. Volume 42, Issue 4, 2009.-pp.269-288.