Int j simul model 16 (2017) 2, 302-312
ISSN 1726-4529 Original scientific paper

A UNIVERSAL CAD SYSTEM FOR CUTTING STOCK PROBLEM

Kovacic, M.[*,**] & Brezocnik, M.[***]
[*] Štore Steel d.o.o., Železarska 3, 3220 Štore, Slovenia
[**] Institute of Metals and Technology, Lepi pot 11, 1000 Ljubljana, Slovenia
[***] University of Maribor, Faculty of Mechanical Engineering, Laboratory for Intelligent Manufacturing Systems, Smetanova 17, 2000 Maribor, Slovenia
E-Mail: miha.kovacic@store-steel.si, miran.brezocnik@um.si

Abstract
In this research the universal system using CAD geometry for solving of cutting stock problem is proposed. The system consists of the three main modules: geometry definition module, objective definition module, and the search strategy module. In the first module the reference points and orientation of parts and stock are defined. In the second module some constraints such as the quantity of the desired type of parts to be placed on the stock can be taken into account. The third module contains the search strategy. For the purpose of this research, the genetic algorithm was used as the search method. For illustration of the generality of the proposed approach, three practical examples for solving orthogonally and irregularly-shaped cutting stock problems are presented. The parts which are to be cut from stock-material with minimal loss and also the stock-material itself are presented as CAD geometries without any kind of geometrical constraints (i.e., concavity, convexity, sections, and holes). In the presented approach the AutoCAD environment was used. The AutoLISP in-house developed system for cutting stock problem was integrated into the AutoCAD. The developed system is without any kind of geometrical constraints and thus highly applicable in the practice. The presented approach can be developed also in other CAD systems where the application programming interfaces (API) are available.

(Received in December 2016, accepted in January 2017. This paper was with the authors 2 weeks for 1 revision.)

Key Words: CAD System, Cutting Stock Problem, Irregular Shapes, Genetic Algorithms, AutoCAD

1. INTRODUCTION

In general, scheduling problems involve efficient usage of resources within selected objectives [1-3]. Most of these problems are geometry-dependent (e.g., facility layout, vehicle routing, packing, or cutting stock problems) [4-9]. Solving them calls for a colourful variability of geometry representation, search methods (i.e., algorithms), and objectives selection.

The geometry-dependent problem solving involves selection of elements (resources), dimensions of shapes (1D, 2D, 3D, or *n*D), and its representation [5, 8-10]:

- 1D (one-dimensional shapes): bars [11];
- 2D (two-dimensional shapes): rectangles, trapezoids, circles, convex, concave and irregular shapes [7, 12-18];
- 3D (three-dimensional shapes) [19, 20];
- *n*D (*n*-dimensional shapes) [21, 22].

Also several objectives can be considered [5, 8-10]:

- element alignment (orthogonal or arbitrary),
- efficient usage of available resources (e.g., number of cuts, scrap percentage) and
- (real world) application constraints [23].

Finally, geometry representation and objectives strongly influence search strategy selection. Several exact and heuristic strategies can be found in the literature [1, 10, 24].

https://doi.org/10.2507/IJSIMM16(2)10.387

Here, only available research dealing cutting stock problems pertaining to irregularly-shaped products are presented.

Arbel [7] developed a system for tree-based representation of packing stock problem for aircraft manufacturers, where several irregular shapes should be cut from the square stock material. Irregular shapes were simplified using column representation. Several cutting strategies are presented: sequence cutting of a single shape, cutting different shapes while finding the most suitable combination and afterwards adjusting in consideration of the stock dimensions or by using previously used cutting patterns. Similarly, the scheduling of cutting was considered. Exact and heuristic algorithms were also used.

Dalalah et al. [10] proposed an approach for solving 2D bin packing problems when cutting non-convex polygonal items out of non-convex stock that is based on their convex boundaries (convex hull optimization) and several feasibility tests in order to avoid their overlapping inside the stock. These tests consist of angle, boundary, point inclusion and polygon intersection test. The approach efficiency was proved by using several published works and known libraries.

In the paper [16], Weng and Kuo proposed general cutting stock problems where 2D CAD geometry, which is transferred into discrete pixels, is used. Also the placing algorithm using small angle rotations is presented. The time consumption is, normally, strongly correlated with the precision of discretization. The proposed system works within the AutoCAD environment.

Burke et al. [17] developed so-called no-fit polygon generation for avoiding overlapping the shapes which are intended for coupling. The concave, convex and polygons with holes can be taken into account. The developed system constructs an envelope around one shape using another shape to be coupled with it. In that way a safe zone is created. Several known benchmark problems were used for presenting the developed system performance.

Anand et al. [18] used a computer vision system in order to transform images of the irregular shapes into polygons. Afterwards, the genetic algorithm was used for layout optimization.

In the paper [25], the authors proposed relocation of the machinery in a steel plant because of the investment in a new continuous rolling mill. The authors were actually solving a Traveling Salesman Problem by means of genetic algorithm method and by considering that the representation of machines can be 2-dimensional or 3-dimensional without any kind of geometrical restrictions. As a result of the layout (machinery) rearrangement, efficiency of material flow in the steel plant increased by 13.58 %.

In the current article the universal approach using CAD geometry for general solving of cutting stock problems is presented. Due to the extensive usage of evolutionary methods (e.g., [25-27]) the authors used genetic algorithm for optimal solution search.

Accordingly, first the system architecture is presented. Afterwards, three selected cases are shown: one with rectangle shapes and stock, a second with irregular shapes and rectangular stock, and a third with irregular shapes and several irregular stock shapes. At the same time the selected search strategy and objectives are presented. In the end conclusions are drawn and guidelines for future work are given.

2. ARCHITECTURE OF THE PROPOSED APPROACH

The proposed approach for cutting stock problem solving is created within the AutoCAD system. The AutoLISP/VisualLISP API is used as the programming language. The dialect of LISP programming language is incorporated into all contemporary AutoCAD systems. The AutoLISP allows the user to access the AutoCAD's internal drawing database and to manipulate graphical entities. This is the key point which was the basis for the development of the proposed system.

2.1 The main idea

The in-house CAD system for cutting stock problem consists of three major modules:

- geometry definition, where geometry, reference points and orientation of shapes and stock are defined,
- objectives definition, where all desired constraints can be taken into account and
- search strategy, where search strategy can be selected and its parameters set.

The principle of the developed universal CAD system for cutting stock problem is presented in Fig. 1. First, as mentioned, the geometry, reference points and orientation of parts (i.e., shapes) and stock are defined. These data are used for a search strategy module which generates and evaluates different solutions according to constraints defined in an objectives definition module. Afterwards the results of the search strategy module are presented. Search strategy can be any strategy that provides a sufficiently good solution in a problem space. For the purpose of this research, we used the genetic algorithm approach.

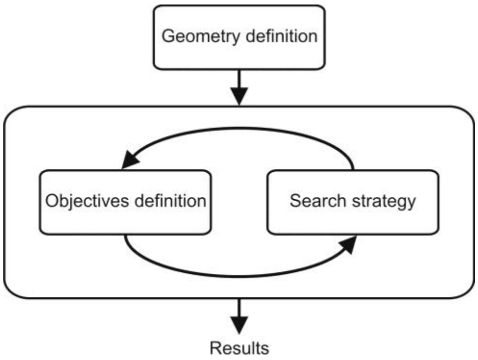


Figure 1: The principle of universal CAD system for cutting stock problem.

2.2 Description of the modules

Fig. 2 shows the possible geometry for products and stock, and their reference points (R1 to R4 for parts/products and Rs for stock) and orientations (o1 to o4 for parts/products and os for stock) of 4 randomly-selected parts and stock. Please note that several stock shapes can be used at the same time. The user can easily select suitable geometry; the points and orientation can be automatically or manually defined.

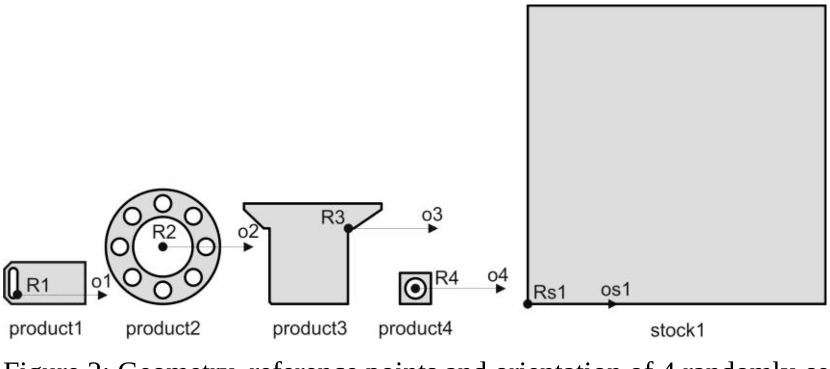


Figure 2: Geometry, reference points and orientation of 4 randomly-selected parts and stock.

It must be emphasized that in a geometry-definition module several geometry properties are also accessible (e.g., area, perimeter, bounding box, mass centre, moments of inertia). Some of them are presented in Fig. 3.

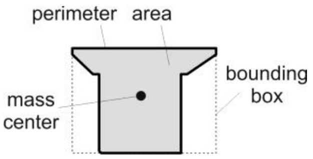


Figure 3: Some accessible geometry properties in the case of *product3*.

In an objectives definition module several constraints, such as number or percentage of required shapes to be cut or needed to be packed, their overlapping area, waste perimeter and area, can be taken into account, Fig. 4.

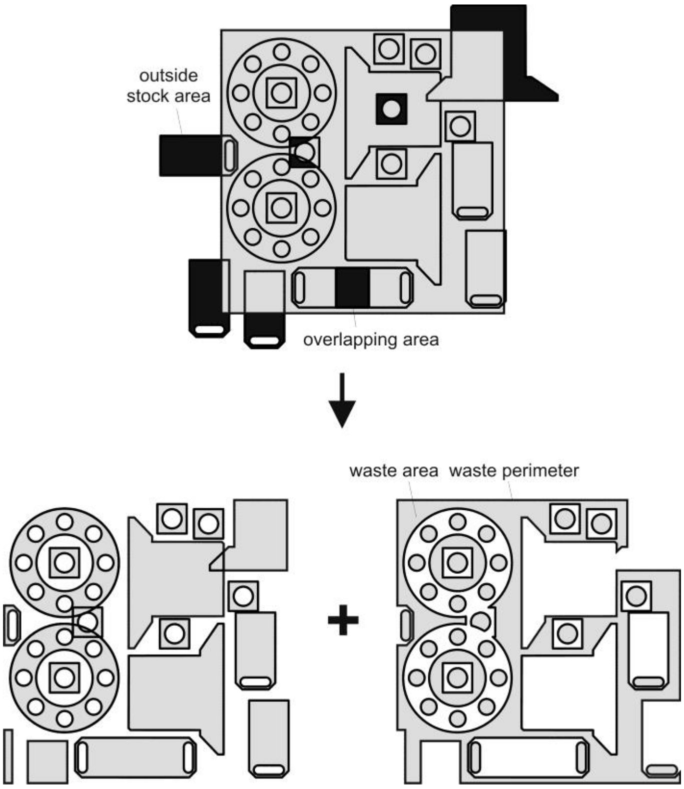


Figure 4: Some accessible geometry properties in an objectives definition module.

In a search strategy module, any kind of search algorithm can be used (e.g., genetic algorithms, particle swarm algorithm, gravitational search algorithm). The input data consists of objectives definition module data (e.g., overlapping area, waste perimeter and area), parts and stock geometry, reference points, and orientation.

3. RESULTS AND DISCUSSION

For this general overview of functionality of the proposed CAD system only cutting stock problems were selected.

The first example shows placing rectangular parts on rectangular stock, the second example shows irregular parts on rectangular stock, and the third shows irregular parts on two irregular stock shapes.

3.1 Definition of objectives

In all three cases presented hereafter the same objectives were selected.

First, the number of required parts of each product should be determined. This can be determined either explicitly by the exact number (e.g., 3 pieces of part 1, 12 pieces of part 2 and 1 piece of part 3) or implicitly (i.e., by the ratio of area of parts of a single product in terms of the area of all parts or of the stock area).

In our case the required ratio between the number of parts was selected. Where there are two products, we can decide that for the first product we need 3 parts and for the second product 6 parts. So the ratio between the number of required parts of the first and the second part is 3 : 6 = 1 : 2. In practice, depending on products and stock dimensions, the number of parts placed on stock can vary. The absolute difference between the mentioned ratio of practically cut and required parts (*fn*) was selected as a first objective.

For the second, third and fourth objectives the ration between waste (*Aw*), overlapping (*Ao*), outside stock (*Aos*) and stock area (*As*) were selected, respectively.

The objective (optimization) function *F* can be defined as:

$$F = w_{fn} \cdot fn + w_{Aw} \cdot \frac{Aw}{As} + w_{Ao} \cdot \frac{Ao}{As} + w_{Aos} \cdot \frac{Aos}{As}, \tag{1}$$

where w_{fn}, w_{Aw}, w_{Ao} and w_{Aos} are corresponding weights which can be adjusted.

In all presented examples the same weights were used: for w_{fn}, w_{Aw}, w_{Ao}, and w_{Aos} we used 1, 1, 20, and 20, respectively. Accordingly, the objective function is defined as:

$$F = 1 \cdot fn + 1 \cdot \frac{Aw}{As} + 20 \cdot \frac{Ao}{As} + 20 \cdot \frac{Aos}{As}. \tag{2}$$

3.2 Search strategy

For the purpose of this research, the genetic algorithms were used as a search strategy. The genetic algorithms are population-based search heuristics which simulate the natural evolution of living beings [28] and can be used for solving different problems (e.g., [29-32]). In general, any other search strategy can be used, such as particle swarm optimization, simulated annealing, bat algorithm, etc. [33-35].

In our case, during simulated evolution the organisms in form of geometric entries and their locations in the stock piece (i.e., solutions to the problem) undergo adaptation. They, through several generations (iterations), gradually change until the stopping criteria are met; this can be the required quality of solution or a maximal number of generations to be run.

Coding of the organisms

The population consists of organisms (i.e., chromosomes). Each individual chromosome in the population represents possible layout of parts in the stock at a population time *t*. As mentioned before, the parts could be of different types. The chromosome consists of genes. Each gene is in fact a part of selected geometric shape. The individual gene consists of the following information: geometry of the part, position of the reference points of the part (i.e., position of the reference points on the stock or even a little bit outside of it), and the orientation of the part. The best chromosome in the population represents the best layout of the parts on the available stock piece.

Genetic operations

For changing organisms, the following eight genetic operations were being created: crossover, small and large position mutation, small and large orientation mutation, permutation, deletion and insertion operation. Also the reproduction operation, which does not change the organisms, is implemented.

The crossover operation ensures the exchanging of genetic material between organisms (Fig. 5).

Position mutation in Fig. 6 a randomly changes the position of a randomly-selected part. The repositioning of this part can be large (i.e., over the stock boundaries) or small (i.e., in the current part vicinity). Similarly, orientation mutation in Fig. 6 b randomly changes the orientation of a randomly-selected part. The changes in orientation can be large (i.e., from 0° to 360°) or small (i.e., within 10°). They can be evenly discrete (e.g. 0°, 90°, 180 and 270°) or arbitrary.

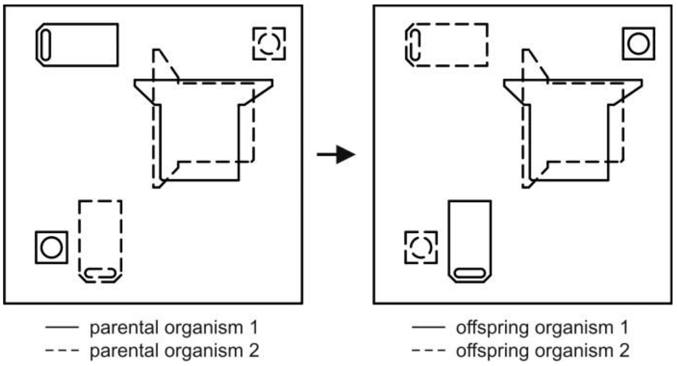


Figure 5: Crossover operation.

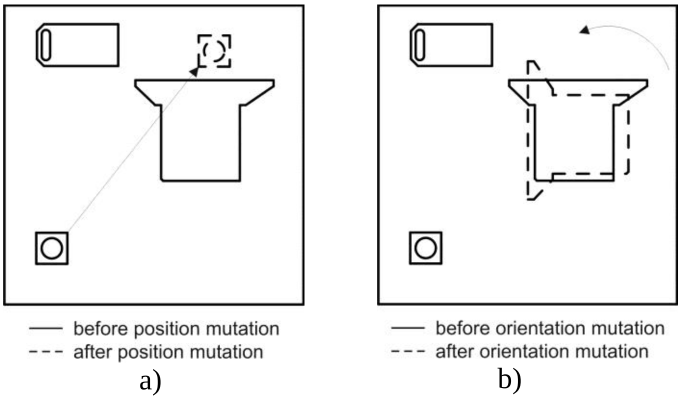

—— before position mutation
--- after position mutation
a)
—— before orientation mutation
--- after orientation mutation
b)

Figure 6: Mutations: a) position mutation; b) orientation mutation.

Permutation operation (Fig. 7) switches the position and orientation between the two randomly-selected parts in the composition.

Deletion operation removes the random part from the composition. The insertion operation adds pars into composition.

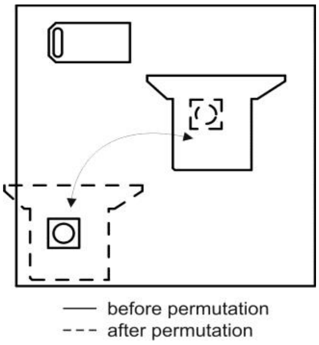


Figure 7: Permutation operation.

Evolutionary parameters

In all experimental runs the same population size of 100 organisms was chosen. The maximum number of generation was 100 for all runs. For selection of organisms, the tournament method with the tournament size of 7 was used. The maximum number of generations was selected as the stopping criteria for the purposes of this paper. The following probability parameters were selected for the simulated evolution control: 0.2 for crossover, 0.2 for large position mutation, 0.2 for orientation mutation 0.1 for permutation, 0.1 for deletion, 0.1 for insertion and 0.1 for reproduction probability. We executed 100 independent runs of the system.

3.3 Example 1

Geometry (parts and stock), reference points and orientation for the first example are presented in Fig. 8. The required ratio between number of parts for each product cut from the stock was 1 : 2 : 2. Due to rectangular shape of the stock and the parts only discrete orientation mutation was used where only 0° and 90° angles were possible.

The results from the most successful run are presented in Fig. 9. For presented organisms (i.e., compositions in generations 0, 50, and 100), the following data are given: the absolute difference between the mentioned ratio of practically cut and required pieces (*fn*), ratios between waste (*Aw*), overlapping (*Ao*), outside stock (*Aos*) and stock area (*As*), and the objective function (*F*).

While observing the results from the generation 0 it is possible to notice that the layout consists of 3, 5 and 6 parts of product1, product2 and product3, respectively. Accordingly, the ratio between number of parts for each product cut from the stock is 3 : 5 : 6 which differs from the required 1 : 2 : 2. According to the definition, the absolute difference between the mentioned ratio of practically cut and required parts (*fn*) is 1 : 2 : 2 – 3 : 5 : 6 = 0.25 – 0.1 = 0.15. The waste (*Aw*), overlapping (*Ao*), and outside the stock area (*Aos*) percentage is 39.64 %, 8.63 % and 13.22 %, respectively. The value of objective (optimization) function *F* (defined in Eq. 2) is:

$$F = 1 \cdot fn + 1 \cdot \frac{Aw}{As} + 20 \cdot \frac{Ao}{As} + 20 \cdot \frac{Aos}{As} \tag{3}$$

$$F = 1 \cdot 0.15 + 1 \cdot 0.3964 + 20 \cdot 0.0863 + 20 \cdot 0.1322 = 4.9180$$

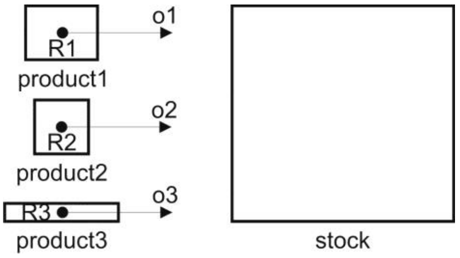


Figure 8: Geometry, reference points and orientations for example 1.

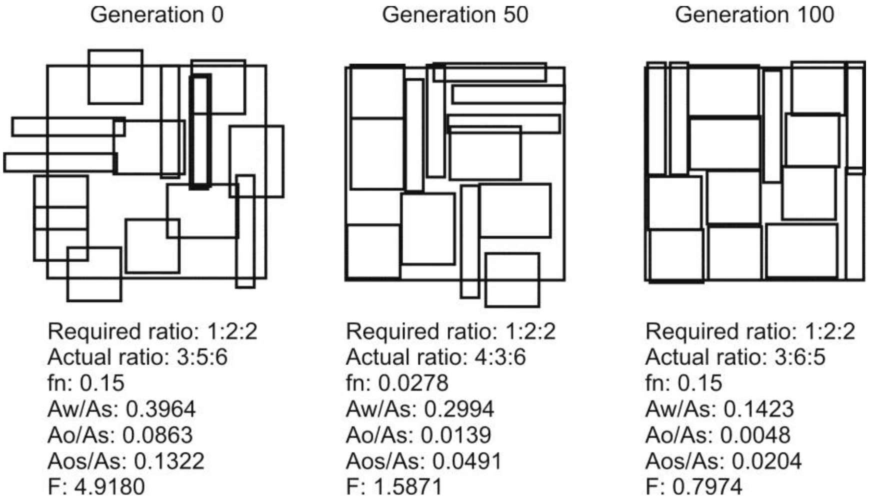
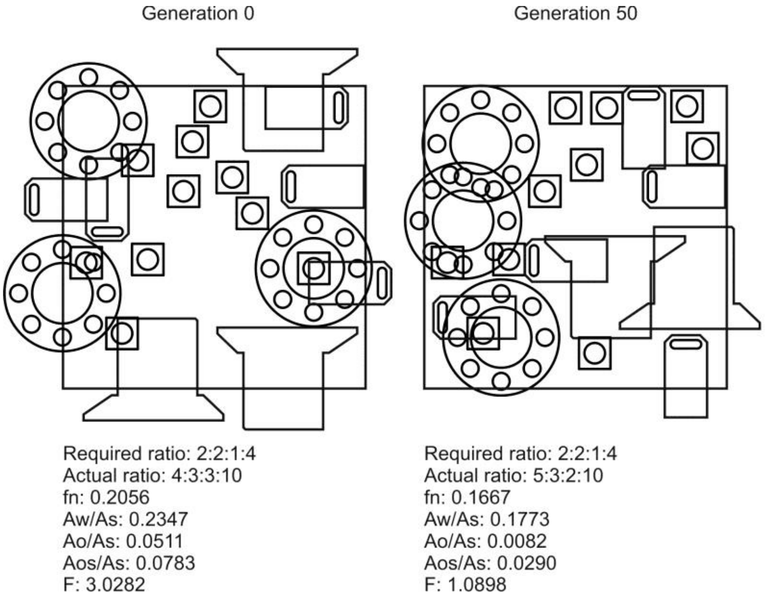


Figure 9: The best organisms – compositions in generations 0, 50 and 100 for example 1.

3.4 Example 2

Geometry (parts and stock), reference points and orientation for the second example are presented in Fig. 2. For this example also discrete orientation mutation was used where 0°, 90°, 180° and 270° angles were possible.

The required ratio between the number of parts for each product cut from the stock is 2 : 2 : 1 : 4. The results from the most successful run are presented in Fig. 10.



Required ratio: 2:2:1:4
Actual ratio: 4:3:3:10
fn: 0.2056
Aw/As: 0.2347
Ao/As: 0.0511
Aos/As: 0.0783
F: 3.0282

Required ratio: 2:2:1:4
Actual ratio: 5:3:2:10
fn: 0.1667
Aw/As: 0.1773
Ao/As: 0.0082
Aos/As: 0.0290
F: 1.0898

Required ratio: 2:2:1:4
Actual ratio: 4:3:2:7
fn: 0.1548
Aw/As: 0.0843
Ao/As: 0.0029
Aos/As: 0.0121
F: 0.5381

Fig. 10: The best organisms – compositions in generations 0, 50 and 100 for example 2.

2.5 Example 3

In this example two stock materials were selected. For demonstration purposes the stock material is coloured grey, Fig. 11. The arbitrary orientation was used (i.e., from 0° to 360°).

The required ratio between the number of parts for each product cut from the stock is 1 : 2 : 6. The results from the most successful run are presented in Fig. 12.

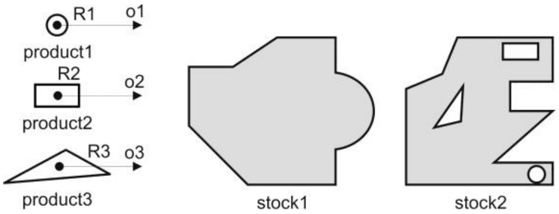
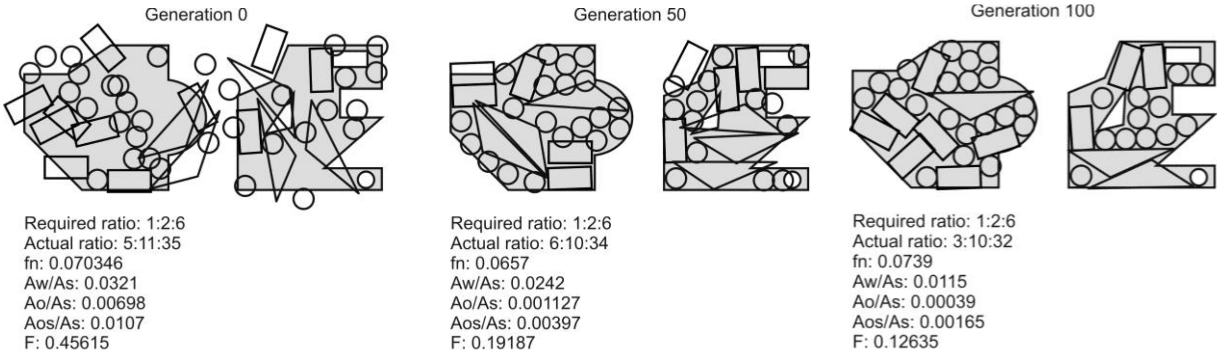


Fig. 11: Geometry, reference points and orientation for example 3.



Required ratio: 1:2:6
Actual ratio: 5:11:35
fn: 0.070346
Aw/As: 0.0321
Ao/As: 0.00698
Aos/As: 0.0107
F: 0.45615

Required ratio: 1:2:6
Actual ratio: 6:10:34
fn: 0.0657
Aw/As: 0.0242
Ao/As: 0.001127
Aos/As: 0.00397
F: 0.19187

Required ratio: 1:2:6
Actual ratio: 3:10:32
fn: 0.0739
Aw/As: 0.0115
Ao/As: 0.00039
Aos/As: 0.00165
F: 0.12635

Fig. 12: The best organisms – compositions in generations 0, 50 and 100 for example 3.

4. CONCLUSION

In this paper the general approach for solving of cutting stock problem is proposed. The in-house system for cutting stock problem solving was realized in the AutoLISP/VisualLISP programming language within the AutoCAD system. The proposed approach consists of three major modules: geometry definition, objectives definition, and search strategy. For the purpose of this research we used genetic algorithms as the search strategy.

Without any kind of geometrical constraints, several objectives (i.e., optimization functions) and search algorithms can be selected for the best solution search. From a practical point of view, only three examples of cutting stock problems were presented. The first example shows placing rectangular parts on rectangular stock, the second, irregular parts on rectangular stock, and the third, irregular parts on two irregular stock shapes.

For changing of organisms (i.e., layout of parts on available stock piece) we used the following genetic operations: reproduction (i.e., selection), crossover, small and large position mutation, small and large orientation mutation, permutation, deletion, and insertion.

In all three experimental cases the difference of the ratios between the number of required and cut pieces, the ratio between waste area, overlapping area between pieces, area of the pieces outside stock and stock area were selected as objectives (objective function). Obviously, minimizing selected objectives is the main aim of the problem. In general also other objectives can be chosen (e.g., waste perimeter).

The results of the best independent run for each of the three cases allow us to conclude that each generation brings improvements in terms of all the selected objectives.

The shortest and longest runs of CAD system for the selected examples on AutoCAD last less than 2 hours and more than 6 hours on an Intel Core i7-4600U processor with 8 GB of RAM, respectively. The majority of the time was spent on manipulation of the geometric entities inside the AutoCAD system.

It must be emphasized that the proposed system is based on CAD geometries without any kind of geometrical constraints (i.e., concavity, convexity, sections and holes) and that it is, with proper objectives and search algorithms selection, extremely applicable in practice. The developed approach will be used also for solving regular and irregular 3D bin packing problem. In addition, in order to shorten the processing time, a fine tuning of the program code will be implemented in the future.

REFERENCES

[1] Pinedo, M. L. (2012). *Scheduling: Theory, Algorithms, and Systems*, 4th ed., Springer, New York, doi:10.1007/978-1-4614-2361-4

[2] Baker, K. R.; Trietsch, D. (2009). *Principles of Sequencing and Scheduling*, John Wiley & Sons, Inc., Hoboken, doi:10.1002/9780470451793

[3] Błażewicz, J.; Ecker, K. H.; Pesch, E.; Schmidt, G.; Węglarz, J. (2007). *Handbook on Scheduling: From Theory to Applications*, Springer, Berlin, doi:10.1007/978-3-540-32220-7

[4] Galzina, V.; Lujić, R.; Šarić, T. (2012). Adaptive fuzzy particle swarm optimization for flow-shop scheduling problem, *Tehnicki vjesnik – Technical Gazette*, Vol. 19, No. 1, 151-157

[5] Cagan, J.; Shimada, K.; Yin, S. (2002). A survey of computational approaches to three-dimensional layout problems, *Computer-Aided Design*, Vol. 34, No. 8, 597-611, doi:10.1016/S0010-4485(01)00109-9

[6] Meller, R. D.; Gau, K.-Y. (1996). The facility layout problem: Recent and emerging trends and perspectives, *Journal of Manufacturing Systems*, Vol. 15, No. 5, 351-366, doi:10.1016/0278-6125(96)84198-7

[7] Arbel, A. (1993). Large-scale optimization methods applied to the cutting stock problem of irregular shapes, *International Journal of Production Research*, Vol. 31, No. 2, 483-500, doi:10.1080/00207549308956738

[8] Han, W.; Bennell, J. A.; Zhao, X.; Song, X. (2013). Construction heuristics for two-dimensional irregular shape bin packing with guillotine constraints, *European Journal of Operational Research*, Vol. 230, No. 3, 495-504, doi:10.1016/j.ejor.2013.04.048

[9] Cheng, C. H.; Feiring, B. R.; Cheng, T. C. E. (1994). The cutting stock problem – A survey, *International Journal of Production Economics*, Vol. 36, No. 3, 291-305, doi:10.1016/0925-5273(94)00045-X

[10] Dalalah, D.; Khrais, S.; Bataineh, K. (2014). Waste minimization in irregular stock cutting, *Journal of Manufacturing Systems*, Vol. 33, No. 1, 27-40, doi:10.1016/j.jmsy.2013.11.003

[11] Corominas, A.; Bautista, J. (1994). Procedures for solving a 1-dimensional cutting problem, *European Journal of Operational Research*, Vol. 77, No. 1, 154-168, doi:10.1016/0377-2217(94)90035-3

[12] Jin, M.; Ge, P.; Ren, P. (2015). A new heuristic algorithm for two-dimensional defective stock guillotine cutting stock problem with multiple stock sizes, *Tehnicki vjesnik – Technical Gazette*, Vol. 22, No. 5, 1107-1116, doi:10.17559/TV-20150731113849

[13] Ouafi, R.; Khelladi, A. (2009). Non orthogonal cutting problem: The case of trapezoidal pieces, *International Journal of Open Problems in Computer Science and Mathematics*, Vol. 2, No. 1, 69-84

[14] Cui, Y.; Chen, F.; Liu, R.; Liu, Y.; Yan, X. (2010). A simple algorithm for generating optimal equal circle cutting patterns with minimum sections, *Advances in Engineering Software*, Vol. 41, No. 2, 401-403, doi:10.1016/j.advengsoft.2009.07.005

[15] George, J. A.; George, J. M.; Lamar, B. W. (1995). Packing different-sized circles into a rectangular container, *European Journal of Operational Research*, Vol. 84, No. 3, 693-712, doi:10.1016/0377-2217(95)00032-L

[16] Weng, W.-C.; Kuo, H.-C. (2011). Irregular stock cutting system based on AutoCAD, *Advances in Engineering Software*, Vol. 42, No. 9, 634-643, doi:10.1016/j.advengsoft.2011.04.009

[17] Burke, E. K.; Hellier, R. S. R.; Kendall, G.; Whitwell, G. (2007). Complete and robust no-fit polygon generation for the irregular stock cutting problem, *European Journal of Operational Research*, Vol. 179, No. 1, 27-49, doi:10.1016/j.ejor.2006.03.011

[18] Anand, S.; McCord, C.; Sharma, R.; Balachander, T. (1999). An integrated machine vision based system for solving the nonconvex cutting stock problem using genetic algorithms, *Journal of Manufacturing Systems*, Vol. 18, No. 6, 396-415, doi:10.1016/S0278-6125(00)87642-6

[19] de Queiroz, T. A.; Miyazawa, F. K.; Wakabayashi, Y.; Xavier, E. C. (2012). Algorithms for 3D guillotine cutting problems: Unbounded knapsack, cutting stock and strip packing, *Computers & Operations Research*, Vol. 39, No. 2, 200-212, doi:10.1016/j.cor.2011.03.011

[20] Gonçalves, J. F.; Resende, M. G. C. (2013). A biased random key genetic algorithm for 2D and 3D bin packing problems, *International Journal of Production Economics*, Vol. 145, No. 2, 500-510, doi:10.1016/j.ijpe.2013.04.019

[21] Gilmore, P. C.; Gomory, R. E. (1965). Multistage cutting stock problems of two and more dimensions, *Operations Research*, Vol. 13, No. 1, 94-120, doi:10.1287/opre.13.1.94

[22] Csirik, J.; van Vliet, A. (1993). An on-line algorithm for multidimensional bin packing, *Operations Research Letters*, Vol. 13, No. 3, 149-158, doi:10.1016/0167-6377(93)90004-Z

[23] Malaguti, E.; Durán, R. M.; Toth, P. (2014). Approaches to real world two-dimensional cutting problems, *Omega*, Vol. 47, 99-115, doi:10.1016/j.omega.2013.08.007

[24] Cintra, G. F.; Miyazawa, F. K.; Wakabayashi, Y.; Xavier, E. C. (2007). A note on the approximability of cutting stock problems, *European Journal of Operational Research*, Vol. 183, No. 3, 1328-1332, doi:10.1016/j.ejor.2005.09.053

[25] Kovačič, M.; Rožej, U.; Brezočnik, M. (2013). Genetic algorithm rolling mill layout optimization, *Materials and Manufacturing Processes*, Vol. 28, No. 7, 783-787, doi:10.1080/10426914.2012.718475

[26] Brezocnik, M.; Kovacic, M.; Ficko, M. (2004). Prediction of surface roughness with genetic programming, *Journal of Materials Processing Technology*, Vol. 157-158, 28-36, doi:10.1016/j.jmatprotec.2004.09.004

[27] Brezocnik, M.; Kovacic, M. (2003). Integrated genetic programming and genetic algorithm approach to predict surface roughness, *Materials and Manufacturing Processes*, Vol. 18, No. 3, 475-491, doi:10.1081/AMP-120022023

[28] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed., Addison-Wesley Longman Publishing Co., Inc., Reading

[29] Hrelja, M.; Klancnik, S.; Irgolic, T.; Paulic, M.; Jurkovic, Z.; Balic, J.; Brezocnik, M. (2014). Particle swarm optimization approach for modelling a turning process, *Advances in Production Engineering & Management*, Vol. 9, No. 1, 21-30, doi:10.14743/apem2014.1.173

[30] Glažar, V.; Perčić, M.; Marunić, G.; Franković, B. (2016). A comparative study of evolutionary and search algorithms for optimization of heat exchanger with microchannel coil, *Transactions of FAMENA*, Vol. 40, No. SI-1, 75-84

[31] Modrák, V.; Pandian, R. S. (2010). Flow shop scheduling algorithm to minimize completion time for *n*-jobs *m*-machines problem, *Tehnicki vjesnik – Technical Gazette*, Vol. 17, No. 3, 273-278

[32] Li, Y.-X.; Yao, X.-F.; Zhou, J.-F. (2016). Multi-objective optimization of cloud manufacturing service composition with cloud-entropy enhanced genetic algorithm, *Strojniski vestnik – Journal of Mechanical Engineering*, Vol. 62, No. 10, 577-590, doi:10.5545/sv-jme.2016.3545

[33] Tang, M.; Gong, D.; Liu, S.; Zhang, H. (2016). Applying multi-phase particle swarm optimization to solve bulk cargo port scheduling problem, *Advances in Production Engineering & Management*, Vol. 11, No. 4, 299-310, doi:10.14743/apem2016.4.228

[34] Marković, D.; Petrović, G.; Ćojbašić, Ž.; Marinković, D. (2012). A comparative analysis of metaheuristic maintenance optimization of refuse collection vehicles using the Taguchi experimental design, *Transactions of FAMENA*, Vol. 36, No. 4, 25-38

[35] Xu, H.; Bao, Z. R.; Zhang, T. (2017). Solving dual flexible job-shop scheduling problem using a bat algorithm, *Advances in Production Engineering & Management*, Vol. 12, No. 1, 5-16, doi:10.14743/apem2017.1.235