2-Dimensional Rectangles-in-Circles Packing and Stock Cutting with Particle Swarm Optimization

Michał Okulewicz
Warsaw University of Technology,
Faculty of Mathematics and Information Science,
Koszykowa 75, 00-662 Warsaw, Poland
M.Okulewicz@mini.pw.edu.pl

***Abstract*—This article aims at closing a gap between the research on solving theoretical packing and stock cutting problems and the research on strictly practical aspects of timber sawing. In order to do that, the article proposes a simple model of a timber log and its sawing procedure, solves a set of packing and stock cutting problems with the usage of a Particle Swarm Optimization based algorithm, and promotes a publicly available benchmark data set for the solved problem.**

In this article a problem of packing a set of rectangular shapes into a circular containers is formulated and solved with the usage of Particle Swarm Optimization algorithm, one-dimensional solutions dictionary and a greedy insertion heuristic for two-dimensional problem. The optimization problem is motivated by the sawn timber production and models the process of cutting out a stock of sawn timber out of the sets of tree logs.

Proposed algorithm provides results of a stable quality with a reasonably high yield (low material waste). Quality of results obtained for different user parameter settings gives also some insight about the trade-off between the value of produced stock, material loss, number of created product types and possible production process costs (which are related to the number of allowed rotations of the input source material).

I. Introduction

Solving packing and stock cutting problems allows for efficient usage of space for storage and usage of input material in producing output resources. Both the packing and stock cutting problems are closely related, but differ in optimization goals. The goal of the packing problem is to find such a coverage of input material with output resource types which will minimize material loss or maximize the output products value. The goal of the stock cutting problem is to produce a demanded number of each type of products. The common part of both optimization problem lies in the problem representation and constraints. Input material and the product types are represented as geometrical objects in $\mathbb{R}^n$. Products need to be fully contained in the source material and must not overlap with each other.

The most popular literature for two dimensional packing and stock cutting problems discusses the rectangular (rectangles-in-rectangles [1]) and circular problems (circles-in-rectangles [2], circles-in-circles [3]). The problem of cutting a rectangular shapes out the circular shapes, which is presented in this paper, is deliberately omitted in the common typology [4] and therefore not available in the two dimensional benchmark generator [5]. On the other hand, considering rectangles-in-circles packing (RICP) as a problem of packing irregular shapes [6] seems to be too general, thus possibly an inefficient approach.

While RICP has not been considered as a packing or stock cutting problem, the problem of optimizing sawn timber production has been discussed in the operations research literature [7]. This paper proposes an abstract model for such a problem and presents a hybrid approach for solving it.

The rest of the paper is organized as follows. Section II defines the model for the 2-Dimensional Rectangles-in-Circles Packing (2DRICP) and Stock Cutting (2DRICSC). Section III describes the PSO algorithm and discusses its possible applications to discrete optimization problems. Section IV describes the proposed PSO-based hybrid algorithm to solve it. Sections V and VI describe the test set, experiments setup and their results. Finally, section VII concludes the paper.

II. 2-Dimensional Rectangles-in-Circles packing and stock cutting with orthogonal guillotine cuts

The 2DRICP and 2DRICSC problems discussed in this paper can be formulated in the following way. Both problems consider a set of types of rectangles R (defined by their width w_i, thickness h_i, value v_i and demanded number d_i) and a set of circles C (defined by their radius r_j and count m_j). The solution of the problems consists of a set of placings $P : R \times C \rightarrow \{X \times Y \times \{horizontal, vertical\}\}^{n_{i,j}}$ for each of the pairs in $R \times C$.

The fact, that possible placings must be achievable only by the orthogonal guillotine cuts, results in creating sets of rectangular *stripes*, with each *stripe* formed by the rectangles of identical thickness. Sets of adjacent perpendicular *stripes* form a *block* (see Fig. 1).

For 2DRICP the following objective functions over the set P of all $p_{i,j}$ placings are considered:

- Waste minimization

$$f_{waste}(P) = \sum_{j=1}^{|C|} m_j \left(\pi {r_j}^2 - \sum_{i=1}^{|R|} n_{i,j} w_i h_i \right) \qquad (1)$$

- Profit maximization

$$f_{profit}(P) = \sum_{j=1}^{|C|} m_j \left(\sum_{i=1}^{|R|} n_{i,j} v_i \right) \qquad (2)$$

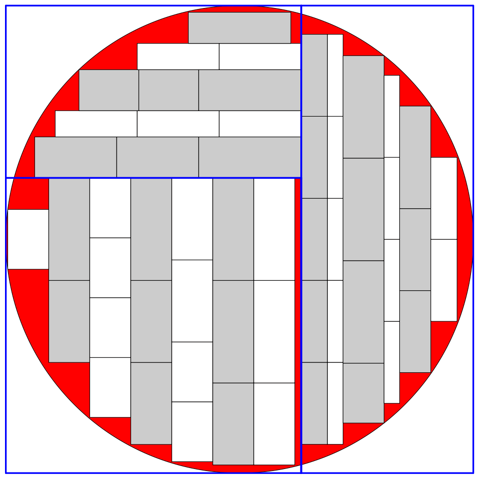


Fig. 1. An example of a circular input material divided into 3 *blocks* (marked by blue borders) with perpendicular *stripes* within each of the *blocks* (depicted with alternating white and gray background).

For 2DRICSC the following objective function over the set P of all $p_{i,j}$ placings is considered:

- Demand fulfilment maximization

$$f_{demand}(P) = \min_{i=1}^{|R|} \frac{\sum_{j=1}^{|C|} m_j n_{i,j}}{d_i} \quad (3)$$

Waste minimization (1) is obtained when the placings have been done in such a way, that a circle is densely packed with rectangular products. Value maximization (2) is obtained for the placings utilizing high number of the most valuable rectangular products, while it may result with a lower input material utilization. Demand fulfilment is obtained, when the value of its objective function (3) is larger then or equal to 1. It is important to note, that while the placings for each circle are independent of each other in the waste minimization and value maximization scenarios, that statement is not true for the demand fulfilment.

The additional observation is, which can be utilized to speed up optimization process, is that a feasible placing for a circle with radius r_k is also a feasible placing for any circle with radius r_l, where $r_l \geq r_k$.

III. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based metaheuristic continuous optimization algorithm, initially designed by Kennedy and Eberhart as an algorithm mimicking simple social behavior [8], quite similar to Reynold's boids [9]. The main idea of the PSO algorithm is to manage a population of particles maintaining their current location x, velocity v (applied to x), memorizing the best visited location (in terms of optimized function) x_{Best} and exchanging information with their neighbors. Further work by Eberhart and Shi [10] resulted in creating the sampling mechanism based on updating the velocity of particles by applying the following rule:

$$\begin{aligned} v \;=\; & \omega v + \quad c_1 u_1 (x_{Best} - x) + \\ & c_2 u_2 (x_{NeighborBest} - x) \end{aligned} \quad (4)$$

Where:

- v is a velocity vector of the particle,
- x is a current location of the particle,
- x_{Best} is a best location visited by the particle,
- $x_{NeighborBest}$ is a best location visited by the neighbors of the particle,

Additionally:

- ω is an inertia coefficient,
- c_1 is a local attraction factor,
- c_2 is a neighbor attraction factor,
- u_1, u_2 are vectors of uniformly distributed random variables.

Apart from the above parameters the communication topology needs to be set, with the typical choice being between: full graph, a random star or a k-ring [11].

Finally, for the discrete problems, it needs to be decided whether PSO will be applied by modifying the algorithm to work in discrete search space [12], [13] or by designing a continuous search space for a given problem [14].

IV. Block cutting PSO-based algorithm

For the problem described in Section II, the following algorithm is proposed. First, a dictionary D of exact 1-D *stripe* solutions is constructed with a dynamic programming approach [15], over all the subsets of rectangles with identical thickness. Subsequently, a division of a circle into a set of *blocks* by cut-and-rotate operation is created. Finally, within each of the *blocks*, perpendicular *stripes* are placed by a greedy algorithm from a dictionary of 1-D solutions.

The 1-D solutions dictionary is iteratively constructed in the following way. Initially the dictionary contains an empty solution with a null value. Possibility of adding new elements to the dictionary is tested, while it is possible to create a new solution maintaining the limit of its maximum length, which is a parameter of the dictionary. In each iteration of the algorithm a single solution from the dictionary is tested for the possibility of enhancement by each of the atomic rectangular products. A new solution is added to the dictionary if there is no solution with higher value consuming less input material. Because the dictionary of solutions would be utilized for finding the most valuable *stripe* within the given length bound applying dynamic programming scheme for finding the dictionary is beneficial not only during construction procedure, but especially during the lookup phase, as all the relevant solutions would be available in the memory.

The search space for the PSO algorithm consists of alternating vertical and horizontal k guillotine cuts coordinates: top-to-bottom, left-to-right, bottom-to-top, right-to-left, top-to-bottom, and so on. Number of cuts k is a user-defined parameter.

TABLE I
SUMMARY OF THE AGGREGATED NUMBER OF TIMBER LOGS' DIAMETERS GENERATED ON THE BASIS OF THE THE HARVESTED TIMBER DESCRIBED IN THE DATA SET [16], CUT IN 2000MM LONG LOGS.

	Diameter	Length	Count
1	(100,300]	2000	6610
2	(300,500]	2000	20702
3	(500,700]	2000	25313
4	(700,900]	2000	20668
5	(900,1100]	2000	13023
6	(1100,1300]	2000	7191
7	(1300,1500]	2000	3510
8	(1500,1700]	2000	555

TABLE II
SAWN TIMBER PRODUCTS' SIZES IN MILLIMETERS, THE VALUE PER THEIR CUBIC METER AND THE DEMAND FOR THE STOCK CUTTING PROBLEM.

	Thickness	Width	Length	Value m^3	Demand
1	19	100	2000	1.00	500000
2	25	100	2000	1.15	500000
3	32	100	2000	1.30	500000
4	38	100	2000	1.41	500000
5	38	125	2000	1.58	500000
6	50	73	2000	1.39	500000
7	50	100	2000	1.62	500000
8	50	125	2000	1.81	500000
9	50	150	2000	1.99	500000
10	50	175	2000	2.15	500000
11	50	200	2000	2.29	500000
12	75	200	2000	2.81	500000
13	75	225	2000	2.98	500000

Greedy solution within each *block* is created by inserting a sequence of *stripes* from D. In each iteration the greedy algorithm chooses a fitting *stripe* with the highest value, with the whole process initialized by selecting a smallest fitting *stripe* on the left or top of the block.

The 2DRICP problem is solved by: (1) creating a common dictionary of 1-D solutions for all of the circles C (limited by the maximum diameter $2r_j$); (2) sampling search spaces of block cuts for each of the circles by an independent PSO optimizers; (3) computing objective function within each block as a sum of values v_i or sizes $w_i h_i$ for rectangles placed by a greedy algorithm.

The 2DRICSC problem is solved by: (1) sampling a search space of all the circles' block cuts by a single PSO optimizer (with solutions computed sequentially in ascending order of their radii); (2) forming a sequence of 1-D solutions dictionaries, separately for each of the circles, with the values of rectangles equal to the demand d_i reduced by the number of rectangles placed in the smaller circles; (3) computing objective function as specified by eq. (3).

V. ASSESSMENT ENVIRONMENT

In order to test the proposed algorithm a set of rectangles has been generated on the basis of the commonly produced sawn pine timber dimensions within the Finnish timber industry. A set of circles has been generated on the basis of statistics of harvested pine trees [16]. The test set is available in the form of JSON files [17] and its summary is presented in Tables I and II.

The test consisted of running the algorithm 10 times, for each of the optimization goals (1)-(3) and 0-6 limit of block cuts. For waste and profit optimization criterion PSO has been run with 40 particles and 500 iterations, while for demand optimization is has been run with 20 particles and 100 iterations. Parameters of the PSO have been set to the following values: $c_1 = c_2 = 1.4$, $\omega = 0.64$, and neighbourhood topology has been set to a random star (with neighbourhood probability $= 0.5$).

The algorithm has been implemented as a single thread application in the .NET Core technology. All of the experiments have been conducted on an Intel Core i5 4300U@1.9GHz processor. Computations time depended on the particular test case. For the whole set of 145 types of input circles computations took from 2.8 seconds for the profit optimization with 0 block cuts up to 24 minutes for 6 cuts for demand fulfilment experiment.

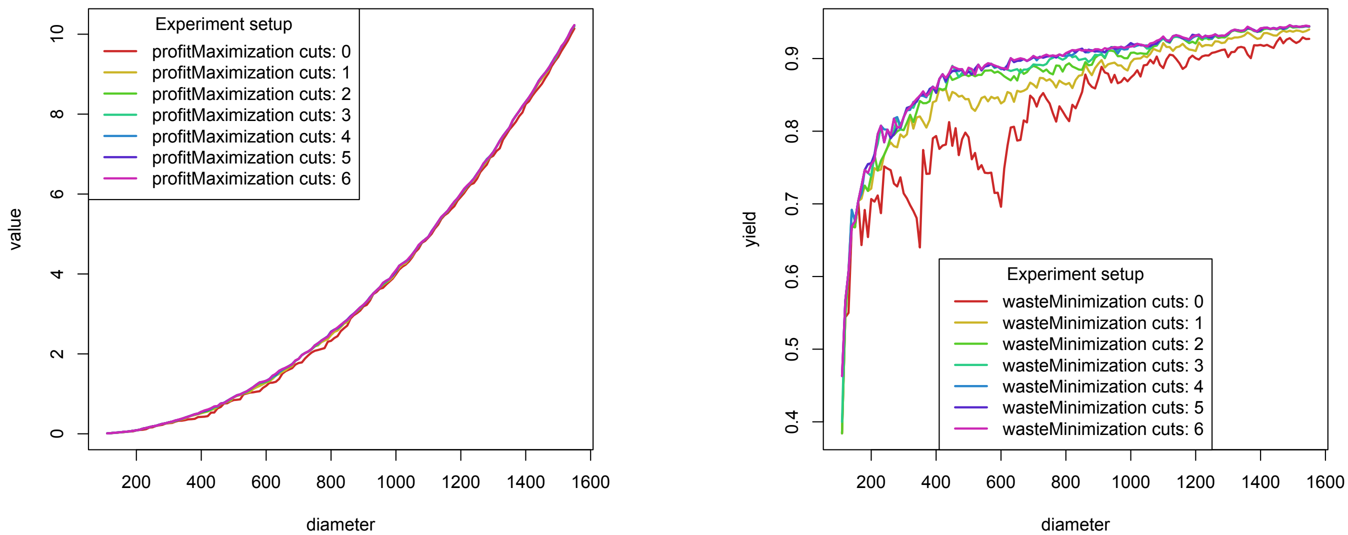
VI. RESULTS

Best obtained numerical results are presented in Tables III and IV. Results present: (a) the total value achieved from all the products, (b) the total material yield (i.e. how much of the input material has been effectively used for creating the output products), (c) the minimum and maximum ratio of the produced number of products to their demanded number over all the 13 types of products. While Tab. III presents the results of packing criteria for all types of experiments, Tab. IV concentrates on presenting the spread of demand fulfilment over various types of products (sawn timber sizes) for the stock cutting problems.

TABLE III
VALUE AND YIELD RESULTS SUMMARY FOR PACKING AND STOCK CUTTING EXPERIMENTS.

	Cuts	Criterion	Value	Yield
1	0	waste minimization	219227	0.8563
2	1	waste minimization	217265	0.8884
3	2	waste minimization	218439	0.9033
4	3	waste minimization	219430	0.9093
5	4	waste minimization	216801	0.9122
6	5	waste minimization	214330	0.9128
7	6	waste minimization	214698	0.9134
8	0	profit maximization	221019	0.8409
9	1	profit maximization	227783	0.8692
10	2	profit maximization	229599	0.8790
11	3	profit maximization	230380	0.8822
12	4	profit maximization	230536	0.8836
13	5	profit maximization	230744	0.8853
14	6	profit maximization	230827	0.8856
15	0	demand fulfillment	180814	0.9141
16	1	demand fulfillment	180818	0.9129
17	2	demand fulfillment	180846	0.9132
18	3	demand fulfillment	180868	0.9133
19	4	demand fulfillment	180703	0.9127
20	5	demand fulfillment	180702	0.9128
21	6	demand fulfillment	180967	0.9142

In order to provide a better insight of the results distribution over various radii of input circles, Fig. 2 and 3 present the



(a) Profit comparison

(b) Material use comparison

Fig. 2. Comparison of the best results for the various number of block cuts over a set of logs' diameters.

comparison for various number of block cuts and distribution of the results for the 6 block cuts.

TABLE IV
RESULTS FOR THE DEMAND FULFILMENT EXPERIMENTS.

	Cuts	Criterion	Min(demand)	Max(demand)
1	0	demand fulfillment	0.9011	1.0177
2	1	demand fulfillment	0.9084	0.9692
3	2	demand fulfillment	0.9090	0.9717
4	3	demand fulfillment	0.9089	0.9701
5	4	demand fulfillment	0.9070	0.9716
6	5	demand fulfillment	0.9087	0.9809
7	6	demand fulfillment	0.9073	0.9769

Figure 2 presents how much can be gained in terms of waste minimization by introducing the possibility of rotating the log (with the lowest line presenting material usage of a greedy heuristic with no cuts). Figure 3 gives insight into the stability of the optimization process by presenting the minimal and maximal obtained values over 10 runs for the 6 block cuts experiment. Diagrams present both the values of the optimization goal within the given experiment and the value of the criterion not used in obtaining the solution.

VII. CONCLUSIONS

It can be observed that, for a given number of block cuts, the optimization process is quite stable, especially for the profit optimization (see Fig. 3). It can be also noted that the number of cuts is a crucial parameter for the waste optimization, while profit and demand results quality are affected to a lesser extent by that parameter (see Tab. III and Fig. 2).

Future work will consist of comparing the computations time with exact algorithm for the smaller examples, and quality comparison with a Monte Carlo game-like approach. Another possible area of research is application of the multiobjective optimization techniques in order to search for Pareto Fronts for all of the optimization criteria. Finally, extending the model in order to solve a 3-D problem is planned.

As a final remark it is worthwhile to observe, that from a production optimization point of view, having a global non-gradient search (i.e. PSO) approach within the optimization procedure gives also one additional advantage over plain heuristic or gradient approaches. Such a non-gradient method makes it possible to add additional information about the production process costs or timber quality estimations to the objective functions values, without redesigning the whole algorithm.

REFERENCES

[1] D. Zhang, L. Shi, S. C. Leung, and T. Wu, "A priority heuristic for the guillotine rectangular packing problem," vol. 116, no. 1, pp. 15–21, 2016. doi: 10.1016/j.ipl.2015.08.008

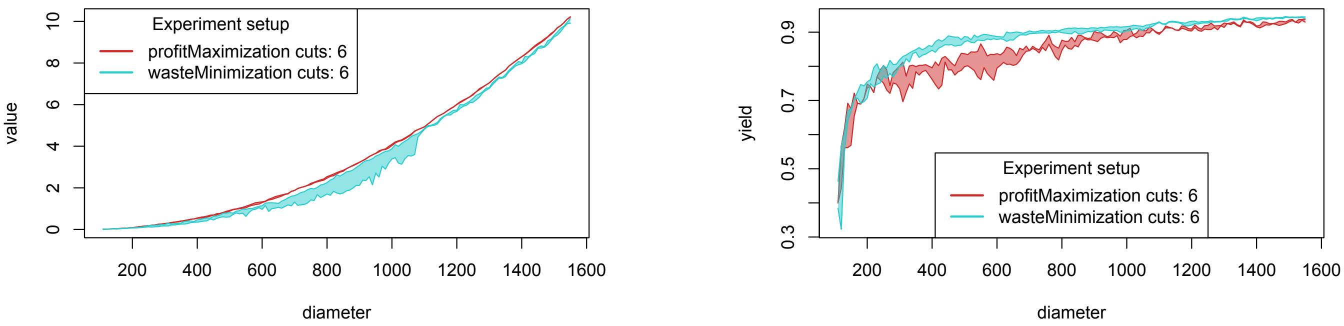
[2] M. Hifi and R. M'Hallah, "Approximate algorithms for constrained circular cutting problems," *Computers & Operations Research*, vol. 31, no. 5, pp. 675–694, apr 2004. doi: 10.1016/S0305-0548(03)00020-0

[3] G. L. Orick, K. Stephenson, and C. Collins, "A linearized circle packing algorithm," *Computational Geometry*, vol. 64, pp. 13–29, 2017. doi: 10.1016/j.comgeo.2017.03.002

[4] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1109–1130, 2007. doi: 10.1016/j.ejor.2005.12.047

[5] E. Silva, J. F. Oliveira, and G. Wäscher, "2DCPackGen : A problem generator for two-dimensional rectangular cutting and packing problems," *European Journal of Operational Research*, vol. 237, no. 3, pp. 846–856, 2014. doi: 10.1016/j.ejor.2014.02.059

[6] A. Martinez-Sykora, R. Alvarez-Valdes, J. Bennell, R. Ruiz, and J. Tamarit, "Matheuristics for the irregular bin packing problem with



(a) Profit comparison

(b) Material use comparison

Fig. 3. Comparison of the results distribution for the highest number of block cuts over a set of logs' diameters.

free rotations," *European Journal of Operational Research*, vol. 258, no. 2, pp. 440–455, 2017. doi: 10.1016/j.ejor.2016.09.043

[7] C. L. Todoroki and E. M. Ronnqvist, "Combined Primary and Secondary Log Breakdown Optimisation," *The Journal of the Operational Research Society*, vol. 50, no. 3, p. 219, mar 1999. doi: 10.2307/3010685

[8] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1. New York, NY, 1995. doi: 10.1109/MHS.1995.494215 pp. 39–43.

[9] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, 1987. doi: 10.1145/37402.37406

[10] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 69–73, 1998. doi: 10.1109/ICEC.1998.699146

[11] M. Clerc, "Standard Particle Swarm Optimization. From 2006 to 2011," 2012. [Online]. Available: http://clerc.maurice.free.fr/pso/SPSO_descriptions.pdf

[12] D. Liu, K. Tan, S. Huang, C. Goh, and W. Ho, "On solving multiobjective bin packing problems using evolutionary particle swarm optimization," *European Journal of Operational Research*, vol. 190, no. 2, pp. 357–382, 2008. doi: 10.1016/j.ejor.2007.06.032

[13] M. R. Khouadjia, E.-G. Talbi, L. Jourdan, B. Sarasola, and E. Alba, "Multi-environmental cooperative parallel metaheuristics for solving dynamic optimization problems," *Journal of Supercomputing*, vol. 63, no. 3, pp. 836–853, 2013. doi: 10.1007/s11227-012-0774-x

[14] M. Okulewicz and J. Mańdziuk, "Two-Phase Multi-Swarm PSO and the Dynamic Vehicle Routing Problem," in *2nd IEEE Symposium on Computational Intelligence for Human-like Intelligence*. Orlando, Fl, USA: IEEE, 2014. doi: 10.1109/CIHLI.2014.7013391 pp. 86–93.

[15] R. Bellman, "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, vol. 60, pp. 503–515, 1954. doi: 10.1090/S0002-9904-1954-09848-8

[16] H. G. Gustavsen, P. Roiko-Jokela, and M. Varmola, *Kivennäismaiden talousmetsien pysyvät (INKA ja TINKA) kokeet: suunnitelmat, mittausmenetelmät ja aineistojen rakenteet (in Finnish)*, ser. Metsäntutkimuslaitos tiedonantoja. Metsäntutkimuslaitos, 1988. [Online]. Available: http://urn.fi/URN:ISBN:951-40-0818-9

[17] M. Okulewicz, "Timber production benchmark set for 2-D Rectangles-in-Circles Packing and Stock Cutting," 2017. [Online]. Available: http://www.mini.pw.edu.pl/~okulewiczm/downloads/lcp/