RESEARCH IN LOGISTICS & PRODUCTION

MULTI-CRITERIA 3-DIMENSION BIN PACKING PROBLEM

Łukasz Kacprzak *, Jarosław Rudy * and Dominik Żelazny *

* Institute of Computer Engineering, Control and Robotic,
Wrocław University of Technology, Wrocława, dolnyśląsk, 50-370, Poland,
Email: lukasz.kacparzak@pwr.edu.pl
Email: jaroslaw.rudy@pwr.edu.pl
Email: dominik.zelazny@pwr.edu.pl

Abstract In this paper a multi-criteria approach to the 3-dimensions bin packing problem is considered. The chosen maximization criteria are the number and the total volume of the boxes loaded into the container. Existing solution representation and decoding method are applied to the problem. Next, two metaheuristic algorithms, namely simulated annealing and genetic algorithm are developed using the TOPSIS method for solution evaluation. Both algorithms are then used to obtain approximations of the Pareto front for a set of benchmarks from the literature. Despite the fact that both criteria work in favor of each other, we managed to obtain multiple solutions in many cases, proving that lesser number of boxes can lead to better utilization of the container volume and vice versa. We also observed, that the genetic algorithms performs slightly better in our test both in the terms of hyper-volume indicator and number of non-dominated solutions.

Paper type: Research Paper

Published online: 31 January 2015
Vol. 5, No. 1, pp. 85-94

ISSN 2083-4942 (Print)
ISSN 2083-4950 (Online)

Keywords: *bin packing problem, multi-criteria, genetic algorithm, simulated annealing, discrete optimization, Pareto efficiency*

1. INTRODUCTION

The field of operations research contains a number of NP-hard discrete optimization problems that are commonly encountered in practical situations like production planning and logistics. In the case of logistics the researchers are often met with a variety of packing problems, the bin packing and container loading problems being the prime examples. Both problems consider a container (also called a bin) and a set of objects (items, boxes etc.). Both the container and the boxes have a certain size (often understood as volume, dimensions or mass). The most common goal is to optimize the number of containers used or pack a single container to its fullest, so as little space is wasted as possible. This problem is connected with the knapsack problem.

The practical applications of the bin packing and container loading problems include situations like loading of tractor trailer tracks, cargo airplanes and ships, as the loading properties considerably affect fuel efficiency, possible transportations strategies (vehicle and roads weight and size limits) as well as vehicle manoeuvrability and thus safety. The aforementioned problems are also found practical use outside of the traditional logistics, i.e. in computer industry. This includes various block packing problems in semiconductor chip design, like Field Programmable Gate Array (FPGA) and Very-Large-Scale Integration (VLSI) circuits, as well as creation and management of backup of files.

Since the formulation of those problems, the most common models assumed a single optimization criterion, such as minimizations of the numbers of bin used, balancing the load of each bin (as close to average as possible), maximizing the number of items loaded or minimizing the space wasted in the container. However, due to the progression of the fields of Multi-Objective Optimization (MOO) and Multi-Criteria Decision Making/Analysis (MCDM/A) in the last years, a number of multi-criteria versions of the discrete optimization problems garnered much attention from researchers around the world. This includes the packing problems with bicriteria approaches, like for example a maximization of both the number of items loaded (more orders fulfilled) and the amount of space used (less space wasted). The multi-criteria container loading problem is the primary focus of this paper, which is organized as follows: in section 2 we present the literature overview and section 3 defines the multi-criteria bin packing problem. Section 4 explains the solution representation and decoding methods used with algorithms described in section 5. The numerical experiments are presented in section 6 and the final sections contains the conclusions.

2. LITERATURE OVERVIEW

Many variations of packing problems have been studied over the years. Lately, researchers have taken into consideration problems with two or more criteria. Multi-criteria optimization has been studied for over two decades, but there is still many research to be done, especially in case of multi-criteria bin packing and container loading problems.

An analysis of solving two-dimensional bin packing problems with rotations and load balancing using parallel and multi-objective memetic algorithms that apply a set of search operators was presented in paper (Fernándeza, Gila, Bañosb & Montoyaa, 2013). Results obtained using a set of test problems show the good performance of parallel and multi-objective memetic algorithms in comparison with other methods found in the literature.

In paper (Dahmania, Clautiauxb, Krichena & Talbib, 2013), authors concurrently optimize three cost functions. They propose two population-based metaheuristics, which use different indirect encoding approaches in order to find good permutations of items which are then packed by a separate decoder routine whose parameters are embedded in the solution encoding. It leads to a self-adaptive metaheuristic where the parameters are adjusted during the search process. The performance of these strategies is assessed and compared against benchmarks inspired from the literature.

A bi-objective 2-dimensional vector packing problem (Mo2-DBPP) that calls for packing a set of items was addressed in paper (Dahmania, Clautiauxb, Krichena & Talbib, 2014). They propose two iterative resolution approaches for solving the Mo2-DBPP, based on the special structure of its Pareto front. Computational experiments are performed on benchmarks inspired from the literature to compare the effectiveness of the two approaches.

A multi-population biased random-key genetic algorithm for the single container loading problem was presented in paper (Gonçalvesa & Resendeb, 2012). The approach uses a maximal-space representation to manage the free spaces in the container. The proposed algorithm hybridizes a novel placement procedure with a multi-population genetic algorithm based on random keys. A novel procedure is developed for joining free spaces in the case where full support from below is required. The computational experiments demonstrate that not only the approach performs very well in all types of instance classes but also it obtains the best overall results when compared with other approaches published in the literature.

In paper (Leung, Wong & Mok, 2008), authors studied a problem of searching for an optimal set of carton boxes for a towel manufacturer so as to lower the overall future distribution costs by improving the carton space utilization and reducing the number of carton types required. A multi-objective genetic algorithm (MOGA) was used to search the optimal design of carton boxes. Clustering techniques were used to study the order pattern of towel products in order to validate the genetically generated results. The results demonstrated that MOGA effectively searched the best carton box spatial design to reduce unfilled space as well as the number of required carton types.

A problem with considered fixed and variable cost, transfer cycle time, flexibility and stacking capacity as the performance indicators was discussed in paper (Golbabaie, Seyedalizadeh Ganji & Arabshahi, 2012). Authors also employed analytical hierarchy process (AHP) to evaluate each alternative layout with respect to each of the criterion.

A multi-objective two-dimensional mathematical model for bin packing problems with multiple constraints (MOBPP-2D) was formulated in (Liu, Tan, Huang, Goh & Ho, 2008). Authors proposed a multi-objective evolutionary particle swarm optimization algorithm, developed to work without the need of combining both objectives into a composite scalar weighting function. Extensive numerical investigations were performed on various test instances, and their performances were compared both quantitatively and statistically with other optimization methods to illustrate the effectiveness and efficiency of the proposed method.

3. PROBLEM DESCRIPTION

The three-dimensional bin packing problem (3D-BPP), in the variant considered in this paper, relies in packing of such a number of boxes into a single container that, with the fulfillment of all adopted restrictions, the total number and volume of boxes loaded is as large as possible.

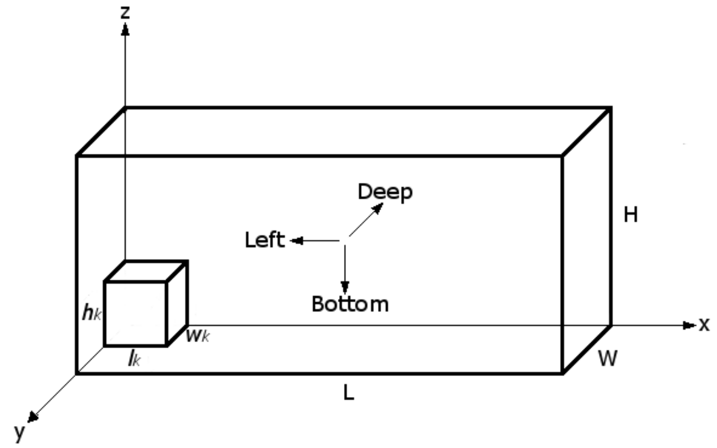

Fig. 1 Container for packing

The container K (see Figure 1), has a form of a rectangular cuboid with fixed dimensions: length L, width W, height H and volume V. There is a set of boxes $P = \{p_1, p_2, \ldots, p_n\}$, type $T = \{t_1, t_2, \ldots, t_m\}$ and rotation $R = \{0, 1, 2, 3, 4, 5\}$, representing possible rotations of boxes. Each box $p_i \in P$ has a certain type u_i, defining its dimensions and possible rotation, i.e. $u_i = t_k$, for $k \in \{1, 2, ..., m\}$, $i = 1, 2, ..., n$, $u \in U = \{u_1, u_2, ..., u_n\}$. A given type $t_k \in T$ can be defined: $t_k = (l_k, w_k, h_k, MR_k)$, where $MR_k \subseteq R$ and determines available rotations, whereas

l_k, w_k, h_k denote respectively: the length, width and height of the box. The set MR_k may contain from one up to six rotations.

A solution of the considered problem defines a set of packed boxes $PZ_e \subseteq P$. The goal is to find feasible solution which maximize both, the number of boxes used and the total volume of the load:

$$\max \sum_{i=1}^{n} c_i \tag{1}$$

$$\max \sum_{i=1}^{n} F_i \tag{2}$$

where

$$c_i = \begin{cases} 1 \text{ if box } \mathrm{p_i} \text{ is used} \\ 0 \text{ otherwise} \end{cases}, \tag{3}$$

$$F_i = c_i v_i , \tag{4}$$

$$v_i = l_i w_i h_i , \tag{5}$$

and meets the restrictions:

1. Each packed box must lie entirely within the container, parallel to the side walls, in one of the available type of rotations,
2. the loaded boxes cannot overlap,
3. the box has to be placed on the bottom of the container, or on top of another.

Coordinates in the Cartesian coordinate system of reference are used to determine the positions of loaded boxes. A set of boxes *P* may vary from weakly heterogeneous to strongly heterogeneous. If the set is weakly heterogeneous, it means that the instance of the problem has few types, with plenty of boxes for each of them, whereas a strongly heterogeneous set consists of many types of boxes, with a small quantity of each type.

4. REPRESENTATION AND DECODING SCHEME

Let us assume that the number of boxes to load is equal to *n*. Each solution is represented by two sets of *n* numbers. The first set describes the types of the boxes, while the second one describes their rotations. For example, in Figure 2 the first box in the representation is of the type 2, while the last one has type 4. The rotations of those boxes are 0 and 3 respectively. It is also possible to store unique numbers of the boxes (instead of their types), but in our case the types and rotations are sufficient description.

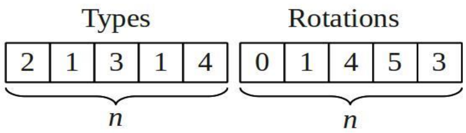

Fig. 2 Example of a solution representation with 5 boxes

In order to determine the values of the objective functions for such representation, a proper decoding method has to be employed. In our case such procedure works as follows. First, the dimensions of all boxes from the representations are altered according to the rotation of each box. Next, the procedure tries to fit each box into the container, starting from the first box. If the procedure concludes that the box cannot be placed in the container, the box in question is simply ignored and next box is processed.

In order to determine whether a box fits in the container or not, the procedure keeps track of a sorted list of available positions (*i.e.* three-point coordinates) for boxes. When an attempt for loading of a box is made, each position is tested and the first feasible position is accepted. At first, the only possible position is $(0,0,0)$, which denotes the beginning of the container. When a box is successfully placed in a container, his chosen position is removed from the list and three new positions are placed there instead. Let (x,y,z) be an accepted position for a box with length l, width w, and height h. Then the new positions have coordinates of $(x+l,y,z)$, $(x,y+w,z)$ and $(x,y,z+h)$. The list of positions is sorted after every successful box loading. The sorting procedure considers all coordinates, with y being the most important and x the least important one.

The last element is a subprocedure responsible for determining whether a given box fits in a given position or not. The box fits only when two conditions are met: 1) box cannot exceed the bounds of the container on any dimension, and 2) second: the box in question does not overlap with any of the already loaded boxes. The second condition requires that the list of already loaded boxes be kept by the decoding method.

Finally, each successfully packed box results in the increase of the values of the object functions: 1 for the number of boxes loaded and lwh for the total volume of the container used. Thus, when the decoding procedure for a given representtation ends, the objective function values are computed.

5. PROPOSED METHODS

For the purpose of this article two approaches, namely Simulated Annealing and Genetic Algorithm, were proposed and implemented. Below we present their background, as well as modifications done in order to adapt those algorithms to the multi-objective bin packing.

5.1. Genetic Algorithm

The Genetic Algorithm (GA) is a population-based metaheuristic, which uses evolution to find better solutions. Evolutionary algorithms use techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover to generate solutions to optimization problems. Usually the evolution starts from random initial population, which is a set of specimens. In each iteration, called a generation, the specimens are modified with the use of genetic operators (namely crossover and mutation) and their fitness is evaluated in order to select best solutions for the next generation.

GA implemented for the purpose of this papers uses Pareto archive, in order to retain non-dominated solutions through successive iterations. Initial population includes random solutions. Mutation is performed by multiple interchanging two random boxes. Crossover randomly cuts two chromosomes and interchanges parts between two individuals. In order to retain feasibility of solutions, boxes are mapped and solutions are repaired. Fitness of each solution is evaluated using TOPSIS technique (Hwang & Yoon, 1981). Tournament selection allows us to choose individuals for next iteration of the algorithm from both populations.

5.2. Simulated Annealing

The SA algorithm belongs to the class of local search methods. Inspired by the phenomenon of metal cooling. With the gradual cooling, the metal particles are distributed in a more systematic way, creating more and more uniform structure. In this process, the cooling temperature is selected so that the particles were divided evenly and found the optimum position. While cooling, before reaching the final temperature, the molecules may adopt a more chaotic position so that, subsequent annealing process steps may occur out of this position to the optimum position.

In each step of the algorithm is executed search the neighborhood of the current solution, during which the accepted solutions are better, and with a certain probability, decreasing with decrease of the temperature, a solution worse than the current one. This allows the algorithm to avoid getting stuck in local optima at the beginning of the search and selecting only better solutions at the end the annealing process. The neighborhood is generated by replacing two boxes and their respective rotations. Current solution is replaced by the newly generated solution, when it is dominated or when the condition is satisfied by the probability of acceptance expressed as $P = \exp((-\Delta) / T)$. In addition, in the course of the algorithm, external archive of non-dominated solutions is built to which are added the new (non-repetitive) Pareto optimal solutions. The algorithm returns an approximation of the Pareto front, instead of a single non-dominated solution.

6. COMPUTER EXPERIMENT

We have performed tests on previously mentioned algorithms and compared them using two multi-objective evaluation techniques. First, for each instance solved by GA and SA algorithms we calculated a number of Pareto solutions (**|P(GA)|** and **|P(SA)|** respectively), as well as number of unique non-dominated solutions from both aggregated approximations of Pareto fronts - **|P|**. Next, we calculated values of Hyper-volume Indicator (**I_H(GA)** and **I_H(SA)** respectively) for each approximation of Pareto front (Zitzler, Brockhoff & Thiele, 2006).

In order to maintain similar test conditions for both algorithms, we have set second stopping condition. Apart from number iterations, we have set a time limit for each instance of the problem. Upon reaching said time limit, algorithms cease their search process and returns approximation of Pareto front, consisting of unique non-dominated solutions extracted from Pareto archive.

Table 1 Values of hyper-volume indicator and numbers of Pareto solutions

No	**Content**											
Instance	**5a**	**5b**	**8a**	**8b**	**10a**	**10b**	**12a**	**12b**	**15a**	**15b**	**20a**	**20b**
|P(GA)|	4	3	2	4	7	3	4	3	5	2	5	4
|P(SA)|	6	2	3	3	4	3	3	4	3	3	3	2
|P|	4	3	3	4	8	3	4	3	6	4	6	5
I_H(GA)	0.70	0.69	0.88	0.70	0.70	0.70	0.69	0.73	0.70	0.67	0.70	0.78
I_H(SA)	0.64	0.67	0.66	0.70	0.74	0.67	0.70	0.64	0.77	0.70	0.67	0.72

Table 1 shows results of our tests performed on 12 instances of bin packing problem. Both algorithms performed well overall, but GA has found more Pareto solutions and had better values of Hyper-volume Indicator in general. It is worth to mention, that neither of tested algorithms fully dominated other one and both had a contribution in final set of non-dominated solutions.

7. CONCLUSION

Multi-criteria optimization allows us to better model complex practical systems than single-criterion approaches, yet some discrete optimization problems still lack research concerning their multi-criteria versions. In this paper we have constructed metaheuristic algorithms for the multi-criteria bin packing problem, a problem that was not extensively considered before in the literature, and showed that the existing representation and decoding methods are sufficient for this task. Moreover, we showed that even while increasing the number of boxes increases the total volume used, it is still possible to obtain higher volume values with lower numbers of boxes. Finally, we notice that the GA algorithm performs slightly better in our research. We conclude that the bin packing is an important part of logistics and using advanced models and algorithms yields considerable results even for the multi-criteria approaches and results in increased competitiveness of companies.

ACKNOWLEDGEMENTS

HUMAN CAPITAL
NATIONAL COHESION STRATEGY

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Project is co-financed by the European Union as part of the European Social Fund.

REFERENCES

Dahmania N., Clautiauxb F., Krichena S., Talbib E.-G., (2013), Iterative approaches for solving a multi-objective 2-dimensional vector packing problem, Computers & Industrial Engineering, Volume 66, Issue 1, pp. 158–170.

Dahmania N., Clautiauxb F., Krichena S., Talbib E.-G., (2014), Self-adaptive metaheuristics for solving a multi-objective 2-dimensional vector packing problem, Applied Soft Computing, Volume 16, pp. 124–136.

Fernándeza A., Gila C., Bañosb R., Montoyaa M.G., (2013), A parallel multi-objective algorithm for two-dimensional bin packing with rotations and load balancing, Expert Systems with Applications, Volume 40, Issue 13, pp. 5169–5180.

Golbabaie F., Seyedalizadeh Ganji S.R., Arabshahi N., (2012), Multi-criteria evaluation of stacking yard configuration, Journal of King Saud University - Science, Volume 24, Issue 1, pp. 39–46.

Gonçalvesa J.F., Resendeb M.G.C., (2012), A parallel multi-population biased random-key genetic algorithm for a container loading problem, Computers & Operations Research, Volume 39, Issue 2, pp. 179–190.

Hwang C.L., Yoon K., (1981), Multiple Attribute Decision Making: Methods and Applications, Springer–Verlag, New York.

Leung S.Y.S., Wong W.K., Mok P.Y., (2008), Multiple-objective genetic optimization of the spatial design for packing and distribution carton boxes, Computers & Industrial Engineering, Volume 54, Issue 4, pp. 889–902.

Liu D.S., Tan K.C., Huang S.Y., Goh C.K., Ho W.K., (2008), On solving multi-objective bin packing problems using evolutionary particle swarm optimization, European Journal of Operational Research, Volume 190, Issue 2, pp. 357–382.

Zitzler E., Brockhoff D., Thiele L., (2006), The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration, Proceedings of EMO 2006, pp. 862-876.

BIOGRAPHICAL NOTES

Łukasz Kacprzak is a Ph.D. Student at the Institute of Computer Engineering, Control and Robotics at Wrocław University of Technology. His research interests include discrete optimization problems, especially bin packing and driver scheduling problem. He is focused on designing good quality approximate parallel methods, which may be addressed to those problems and to determine theirs theoretic properties. He is also interested in advanced computing environments, such as clusters and GPU. He teaches classes related to the area of information technologies. He is the author and co-author of papers which appeared in *Innovations in Management and Production Engineering* conference.

Jarosław Rudy is a Ph.D. Student at Wrocław University of Technology where he received his M.Sc. in computer science and an works as an assistant at the Institute of Computer Engineering, Control and Robotics at the same university. His teaching classes focus on operating systems and programming languages. His research interests include discrete and multi-criteria optimization (scheduling in particular), operations research, computability theory and models of computation. He is the author and co-author of several papers, some of which appeared in *Journal of Applied and Theoretical Computer Science*, *Journal of Applied Computer Science* and in the *Web of Science*.

Dominik Żelazny is a Ph.D. student at Wrocław University of Technology. He teaches discrete optimization, information technology and control of production processes. His research interests are scheduling problems, multi-objective optimization, vehicle routing problems, multi-criteria deciscion making and other discrete optimization problems. He is the co-autor of chapter in book monography "*Discrete optimization in Computer Engineering, Control and Robotics*". His paper appear in numerous conferences including *International Conference on Computational Science*, *International Conference on ICT Management for Global Competitiveness & Economic Growth in Emerging Economies*, *Production engineering: innovations and technologies of the future*, *Innovations in Management & Production Engineering* and *National Conference of Descrite Processes Automation*. His papers appeared in journal *Procedia Computer Science*.