

A Hybrid Methodology Approach for Container Loading Problem Using Genetic Algorithm to Maximize the Weight Distribution of Cargo

Luiz Jonatã Pires de Araújo and Plácido Rogério Pinheiro
*University of Fortaleza (UNIFOR) - Graduate Program in Applied Informatics
 Fortaleza (CE),
 Brazil*

1. Introduction

It is an agreement that maritime transport of goods occupies an important role in economic development throughout the history. For centuries, port cities were in center of economy, where there were traffic of all kind of products and concentration of industrial factories.

In this background, ship containerization brought great advantages to this process. Its invention in mid-1950s was a key factor in development of modern global commerce by bringing down the cost of transporting and reducing time it takes for loading and unloading cargo (Levinson, 2008).

However, the efficient use of containerization involves new and specialized logistic process, a number of technologies, logistics plans and automated systems to handle a great number of containers. To answer these requirements, computation appears as important tool. For example, software can "determine the order in which the containers are to be discharged, to sped the process without destabilizing the ship" (Levinson, 2008).

The described scenario has been treated in academic literature as the Container Loading Problem (CLP), which was firstly approached by Gilmore and Gomory (Gilmore & Gomory, 1965a). There are some variances of this problem in literature and we approach the Knapsack Loading Problem (3D-KLP), that is, the task of to orthogonally pack a subset of given boxes of various sizes within a single container with fixed dimensions optimizing a criterion such as the total loaded volume.

Still according Dyckhoff (Dyckhoff, 1990) and Wascher (Wäscher et al., 2007), the CLP is a NP-hard problem in the strong sense and belongs to cutting and packing problems problem class. It means there is no known polynomial algorithm that exactly solves the CLP in acceptable execution time.

So, to the described problem, specifically the Knapsack Loading Problem (3D-KLP), this work presents a novel backtracking heuristic that not only maximizes the packed cargo volume but also optimizes its weight distribution. It is the great contribution of present work. Mainly if we consider that the cargo to be packed is composed by items with different densities, which turns the problem more difficult. On the other hand, if we are stowing cargoes of uniform

density, weight distribution is not a problem. The figure 1 illustrates a container which mass center of cargo is not necessarily near to its geometric center.

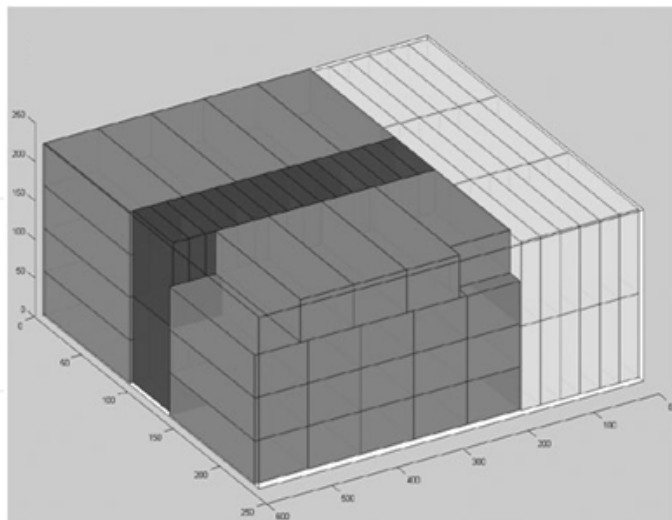


Fig. 1. Different types of packed boxes.

The present methodology is composed by two phases with distinct goals. The first phase is concerned with maximizing the use of container, consequently minimizing the waste of space. It is made by combining a search algorithm, the backtracking, with heuristics that solve integer linear programming models to pack boxes. The second phase executes a Genetic Algorithm to maximize the weight distribution of previously packed cargo. In this work we intend to focus on second phase and consider why genetic algorithm is good alternative to be combined with Heuristics Backtracking.

This work is organized as follows. In section 2 we discuss some related work to solve the CLP, presenting the three main classes of methodologies to approach the problem. In section 3 we present the two phases of our proposed algorithm. We focus in second phase and how we apply a standard genetic algorithm to optimize the weight distribution. Finally, in section 4 we present some computational results. So, in section 5 we make some conclusions regarding the quality of the solutions provided and make some considerations concerning future development.

2. Approaches to solve the container loading problem

In this chapter we present three categories of methodologies existing in theoretical literature on Container Loading Problem: Exact Methods, Heuristics and Hybrid Methods. We discuss these categories and some related work for each category.

2.1 Exact methods

The Container Loading Problem can be modeled as an integer programming problem, as described by Chen (Chen et al., 1993). The proposed model was validated by solving small

problems with up to 6 boxes. In other work, Gilmore and Gomory (Gilmore & Gomory, 1963; 1965b) present another {0-1} integer model to solve the CLP using the Simplex method. In this model, the possible coordinates for place boxes belongs to a discrete set and there are {0-1} decision variables to determine if a box is placed is a specific position and other ones to avoid box overlapping. The model in (Gilmore & Gomory, 1963; 1965b) can be unfeasible due to its large number of variables and constraints. Larger the container is higher the number of variables and constraints.

Some characteristics of methodologies that use only exact methods:

- They aim to find the optimal solution;
- They require higher-level computational resources
- Feasible only for small instances.

Due to cited characteristics, few work exist using only exact methods. It is necessary another methodology more feasible that allow us to find good results in acceptable time.

2.2 Heuristics

However exact methods find the best solution, it becomes impractical due to the necessity of high computational resources. Bypassing this problem, much work proposed strategies, heuristics, to avoid applying exact methods. Now we present some known applications in literature.

One of the most known is the *wall building heuristic*. It was firstly described by George and Robinson (George & Robinson, 1980) to create layers across the depth of the container. Each layer is filled in a number of horizontal strips and their dimensions are determined by the first box, taken from a priority queue. A two-dimensional packing algorithm arranges the boxes within the layers. This heuristic can be effortlessly adapted to build horizontal slices.

Many approaches to CLP are based on 'wall-building'. For example, Pisinger (Pisinger, 2002), that presents an algorithm in which the set of layer dimensions are defined through a backtracking algorithm in order to achieve better results. The wall building strategy was also combined with others methods to attend additional requirements. For example Davies and Bischoff (Davies & Bischoff, 1999) build segments, which are composed of one or more walls, which could be rotated and/or interchanged in order to improve the weight distribution.

It is also possible approach the CLP through metaheuristics, that is, computational methods that make few or no assumptions about the problem to be optimized, and try achieve candidate solutions with good measure of quality. These methods, however achieve good solutions in reasonable time, they do not guarantee the optimal solution. In literature we find some works which apply tabu search or simulated annealing in their algorithms with significant improvements.

Genetic Algorithm (GA) has been successfully used to solve the Container Loading Problem. For example, Gehring (Gehring & Bortfeldt, 1997) reduced the 3D-KLP to two-dimension packing problems by arranging items in stacks to the top of container, a strategy based on (Gilmore & Gomory, 1965a). So, a packing sequence of the stacks is represented as a chromosome. The GA process a population of solutions in order to find a good solution, that is, a good packing plan.

Although heuristics methods have better execution time when compared with exact methods, they do not guarantee to find an optimal solution.

2.3 Hybrid methods

We increasingly find papers that seek to combine exact algorithms and metaheuristics to solve combinatorial optimization problems. Conform Dumitrescu and Stuetzle (Dumitrescu & Stuetzle, 2003), these ideas fall under a category of algorithm that has been commonly referred to as *hybrid methods*.

Nepomuceno et al. (Nepomuceno et al., 2007) introduced a successful work in which reduced instances of 3D-KLP are generated by a genetic algorithm, and then solved by linear programming.

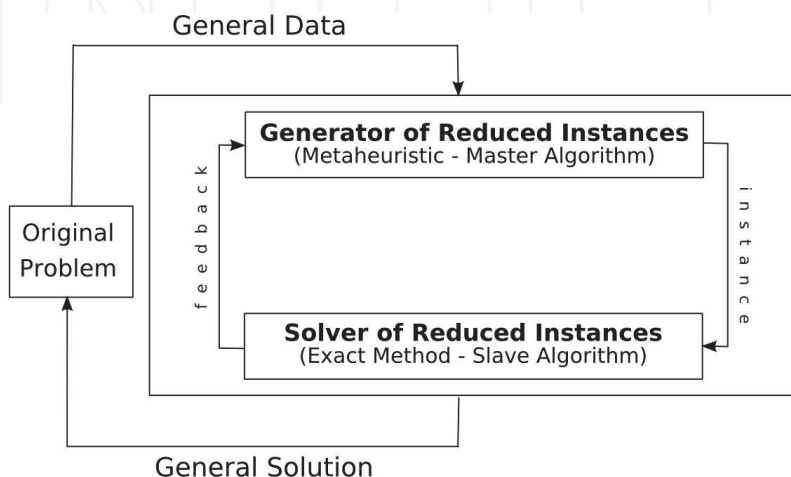


Fig. 2. Flow chart presented in (Nepomuceno et al., 2007).

We also find in literature approaches that combine heuristics methods and local search methods. For example Peng et al. (Peng et al., 2009) combine a basic heuristic algorithm to generate feasible solution from a packing sequence and a search algorithm to find an approximated optimal solution from generated solution.

Thus, once we briefly presented exact, heuristics and hybrids methods and some examples, it is interesting to say that there is no single approach that works better for all problem types or instances. As stated in no free lunch theorem for search and optimization (Wolpert & Macready, 1997), each algorithm is better for a set of specific cases or problem instances while it is worse for other ones.

3. The methodology for 3D knapsack loading problems

As presented in previous works (Araújo, 2011; Araújo & Pinheiro, 2010a;b), the Heuristic Backtracking methodology consists of two independent steps that we call 'phases'. In the first phase, the algorithm is concerned with maximizing the packed volume by combining wall building heuristics with a backtracking search algorithm to choose the best order of proceeding implemented heuristics. In second phase, the algorithm optimizes the weight

distribution of cargo by using a classical genetic algorithm to determine a good arrangement of walls, layers and blocks that were built in first phase. This searching for good arrangements does not affect the packed items by previous phase.

In present work we focus the second phase and the justification in we used genetic algorithm in our approach.

3.1 Phase 1 - Heuristics backtracking

In this phase, we are concerned with maximizing the total volume of packed cargo. It is based on Pisinger's approach (Pisinger, 2000) which combines wall building heuristic with backtracking in order to determine the best dimensions of layers. In other hand, we used backtracking to determine the best implemented heuristics to be used although the packing process for each subproblem. The result is a build tree solution.

The Heuristics Backtracking recursively fill the container creating blocks of boxes that are through proposed heuristics. They are in order: XZ Mixed Layer (a), XY Mixed Layer (b), ZY Mixed Layer (c), Partition on X (d), Partition on Z (e), Partition on XZ - Stack (f), Strip Block on X (g) and Strip Block on Z (h). Examples of built blocks are illustrated in figure 3.

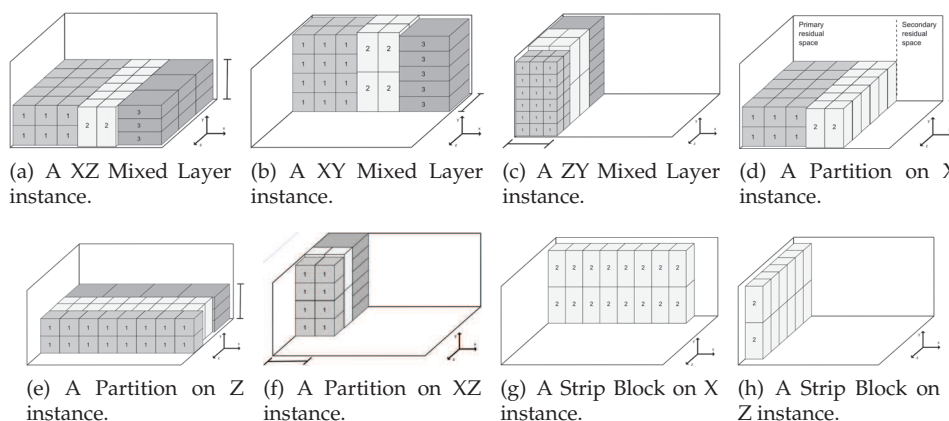


Fig. 3. The used heuristics in the algorithm.

Each heuristic solves a specific integer programming model that aims to maximize the total packed relevance, a non-linear coefficient associated to each box to priorities the bigger ones during packing. The adopted relevance value for i^{th} greatest box type in a list of n box types is $r(i, n) = 2^{n-i}$. Therefore, the algorithm lets the smaller boxes to pack in residual space, when it is small. If the model has solution, the algorithm makes the packing and generates the output problem that will be the input problem for the next recursive call to fill the residual space.

As another characteristic, each heuristic accept a small waste of space. This characteristic allows us to find out good solutions that would be discarded if we would accept only optimal solutions.

We use a tree data structure to maintain the solution where each node keeps the received subproblem (input problem), the well-succeeded heuristic to solve it, the list of packed boxes

heuristics can be rotated as illustrated in figure 5. In other words, each built block of boxes can be arranged in either *even rotation* (no rotation) or *odd rotation*.

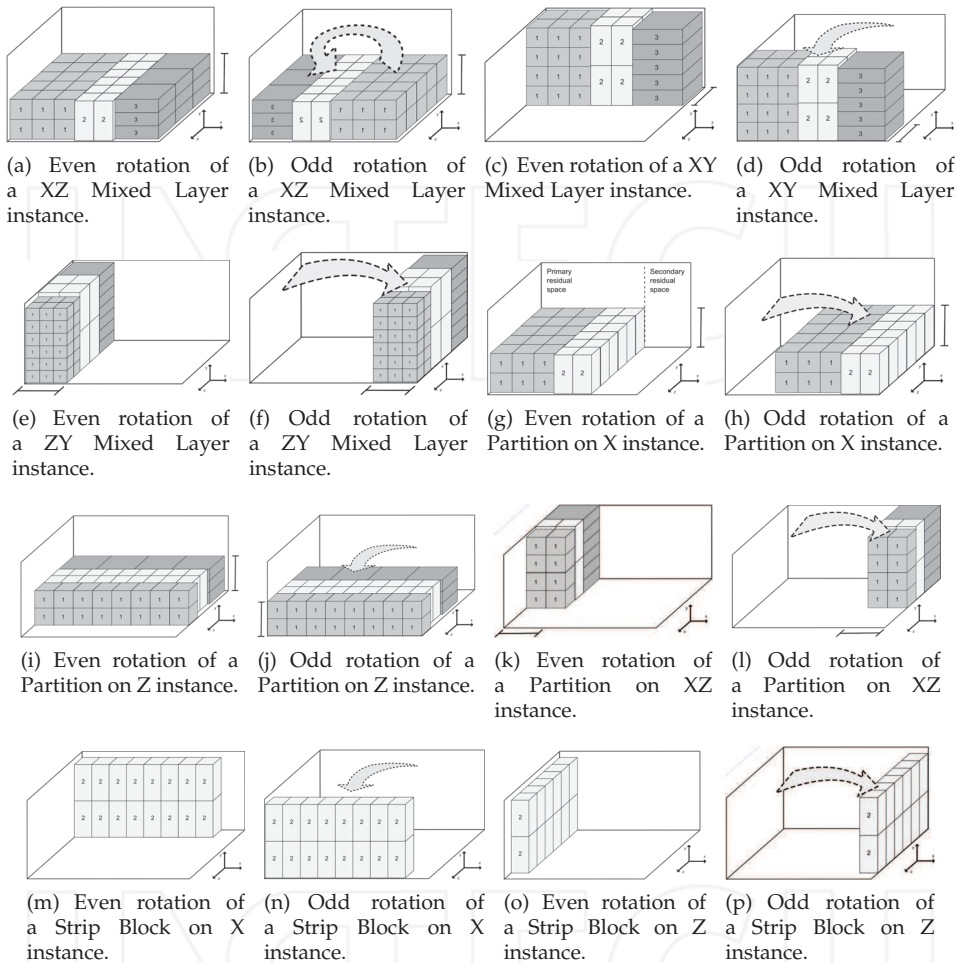


Fig. 5. Rotations of heuristics.

From what was defined, it is possible to represent a tree solution with n significant nodes by a binary chromosome with n genes, a gene g_i for each significant node n_i . The statement $g_i = 0$ indicates the block of boxes in n_i should be arranged in *even rotation*. If $g_i = 1$, in the way called *odd rotation*.

To illustrate how the algorithm changes the weight distribution of a tree solution, we chose that one in figure 4-b, with 5 significant nodes. We have in figure 6 two binary chromosomes with 5 genes that differ in value of node 5. It means the algorithm applies rotation to the block of boxes in node 5 of tree solution, which in turn was built through 'XY Mixed Layer', and consequently resulting in a new weight distribution of cargo.

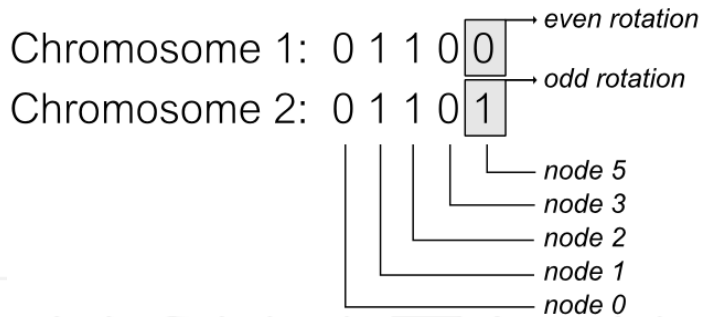


Fig. 6. Different chromosomes obtained from a same solution.

For each individual we can apply an evaluation function that informs us how well the weight distribution is. The proposed evaluation function measures the distance between the geometric center of container and the cargo's center of mass. In both points, y-axis is disregarded. The functions is defined in function 1.

$$f = \sqrt{(x_{gc} - x_{cm})^2 + (z_{gc} - z_{cm})^2} \quad (1)$$

$$x_{cm} = \frac{\sum_{i \in B} x_i w_i}{\sum_{i \in B} w_i}; \quad z_{cm} = \frac{\sum_{i \in B} z_i w_i}{\sum_{i \in B} w_i} \quad (2)$$

where (x_{gc}, z_{gc}) is the geometric center of the container, (x_{cm}, z_{cm}) is the center of mass of the entire cargo, (x_i, z_i) is the center of mass of a box i and w_i its weigh. During implementation we assumed the center of mass as been the geometric center of box.

It is important to note that to calculate the new center of mass from combining the found solution in first phase and a set of rotations described by a chromosome is a very quick operation once it does not change the relative positions of block of boxes. Therefore, it is not necessary to execute again the first phase of implementation. The spent time to evaluate a solution (individual) directly affects the execution time. In a hypothetical scenario, if it necessary 10 seconds to evaluate an individual, a population of 100 individuals that is improved by 60 generations, for example, it leads to a total execution time greater than 16 hours.

In implementation of Genetic Algorithm, which simplified flow cart is illustrated in figure 7, we used its canonical definition:

- Fixed-length binary chromosomes: this size is determined by the quantity of significant nodes of found solution in phase 1;
- Positive fitness domain: once the evaluation is the measuring of the distance between two points, the geometric center of container and the center of mass of cargo, its values is non-negative;
- Fitness proportional selection: the probability p_i of an individual i to be selected to reproduce is $p_i = f_i / \sum_{j=0}^N f_j$, where N is the number of individuals in the population and f_i is the evaluation of individual i ;
- One-point point crossover.

In next sections we propose a new methodology to compare results from different instances and why we chose genetic algorithm to be used with Heuristics Backtracking.

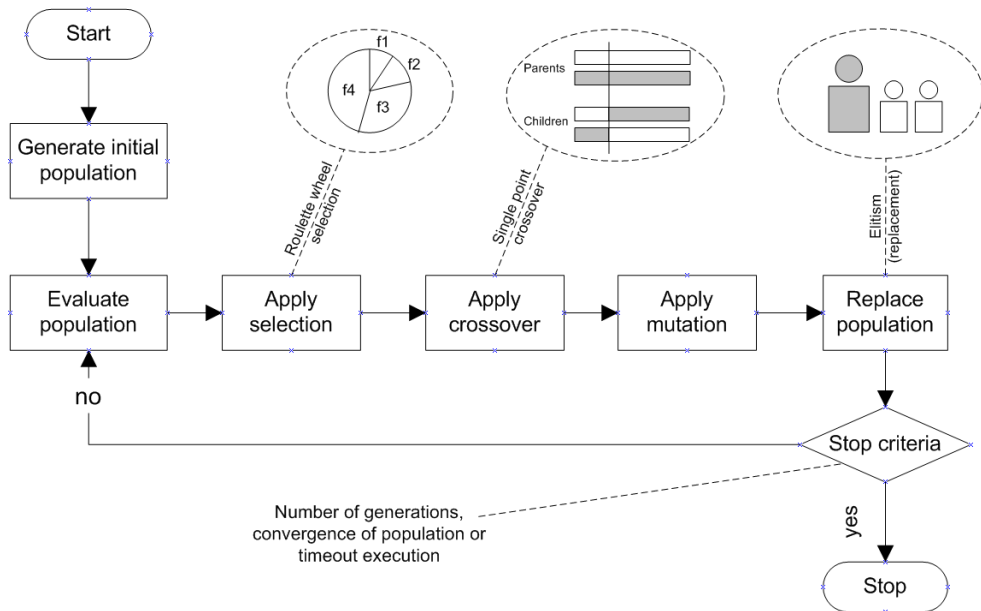


Fig. 7. Simplified flow chart of a Genetic Algorithm.

3.2.1 An approach to compare results regarding weight distribution

There are some known benchmarks in literature to packing problems, including the 3D Knapsack Loading Problem. For example, Bischoff and Ratcliff (Bischoff & Ratcliff, 1995b) proposed a test data for comparing the different approaches for cutting and packing problems. This benchmark and others are cited in section 4.

However, in most of available problems there is no information about weight. In these cases, we can stipulate a formula to define the weight from box dimensions. Moreover, most of these benchmarks differ in many aspects, for example the size of container. In this section we standard the measure to weight distribution requirement in order to facility the comparing between our results and future developments or works, even using different benchmarks.

We stated in previous section about the evaluate function of a solution. It is measured by the distance $d(s)$ between the geometric center of container and the center of mass of the entire cargo in solution s , unconsidered y -axis.

Let it be $Diag$ the length of the diagonal of container floor, where $Diag = \sqrt{W^2 + D^2}$, W and D the width and depth of container, respectively. Thus, we calculate the quality of weight distribution q_{wd} in a solution as follows:

$$q_{wd}(s) = \frac{d(s)}{Diag}$$

To better understand the used variables, they are illustrated in figure 8.

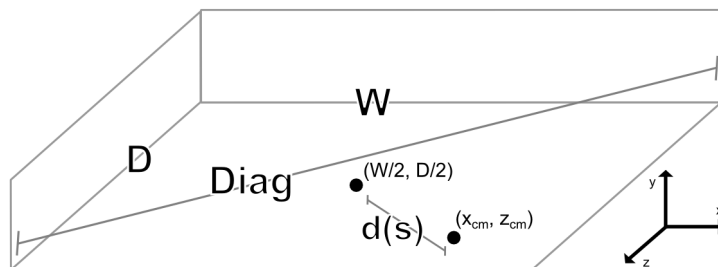


Fig. 8. Variables to measure the quality of the weight distribution of a solution s .

Therefore, in section 4, we will present the efficiency of using of container and the quality considering the weight distribution.

In this section, we presented the two phases of proposed methodology including an approach to compare results concerning weight distribution. In next section we present some computational results to prove our approach's efficiency.

4. Computational results

In order to check the quality of generated solutions, we tested some known benchmark test suites that were found in literature. We present computational results for both kinds of problems, two and three-dimensional ones. We also present a case study in which we compared our results with those in use within ESMALTEC, a stove and refrigerator manufacturer in Brazil.

The computational results were obtained using an Intel Core 2 Duo 2.1 GHz with 3 GB of RAM. The operating system is Windows Vista Home Edition. The development platform is Java 6.0 and Eclipse 3.1.1 tool, while the solver utilized was CPLEX 9.0.

Before present results, it is necessary to notice that container loading instances can be classified according its cargo. According to Dyckhoff (Dyckhoff, 1990) and (Wäscher et al., 2007) the cargo can be regarded as *homogeneous* (identical small box types), *weakly heterogeneous* (many items of relatively few different box types) or *strongly heterogeneous* (many items of many different box types). In order to detect in which type of cargo our approach works better, we tested different kinds of problems, as illustrated in figure 9. Instances of CLP are also classified according to the diversity of box types that can be loaded into the container. We evaluate our proposed algorithm on both homogeneous and weakly heterogeneous instances.

In this sense, Bischoff and Ratcliff (Bischoff & Ratcliff, 1995b) proposed a test data for comparing the different approaches for cutting and packing problems. The benchmark library with 700 problems is available on the internet on the website <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>. This collection of test data sets is used to perform a variety of Operational Research approaches. These data sets contain several types of problems, from homogeneous (BR1) to strongly heterogeneous (BR7). In brackets the number of types of boxes. We compare our obtained results average with the following approaches: a genetic algorithm proposed in (Gehring & Bortfeldt, 1997) (GA), the constructive algorithm on (Bischoff & Ratcliff, 1995b)(CA), a constructive algorithm by Bischoff (Bischoff et al., 1995)(HBal), the hybrid genetic algorithm described in (Bortfeldt & Gehring, 2001)(HGA),

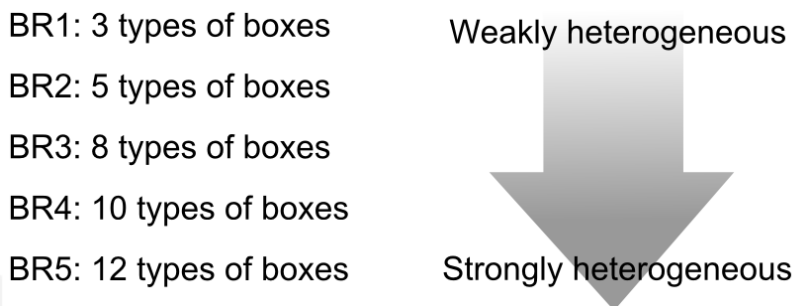


Fig. 9. Types of instances according to the quantity of boxes.

the parallel genetic algorithm in (Gehring & Bortfeldt, 2002)(PGA), a proposed heuristic in (Bischoff, 2006)(PH) and Tabu Search proposed in (Bortfeldt et al., 2003)(TS).

The achieved average results and the standard deviations (σ) by our methodology are represented in columns (HB+GA) and σ , respectively. These results were obtained with timeout execution parameter equals to 10 minutes.

Due to software limitations, for example the maximum number of variables, we tested the first five libraries (BR1 - BR5).

Group	GA	CA	HBal	HGA	PGA	PH	TS	HB+GA	σ
BR1 (3)	86.77	83.37	81.76	87.81	88.10	89.39	93.23	92.13	4.07
BR2 (5)	88.12	83.57	81.70	89.40	89.56	90.26	93.27	91.09	5.09
BR3 (8)	88.87	83.59	82.98	90.48	90.77	91.08	92.86	90.38	8.45
BR4 (10)	88.68	84.16	82.60	90.63	91.03	90.90	92.40	89.07	7.08
BR5 (12)	88.78	83.89	82.76	90.73	91.23	91.05	91.61	88.17	7.52
Average	88.24	83.72	82.36	89.81	90.14	90.54	92.67	90.17	6.44

Table 1. Comparing some proposals, in %.

An instance of solution of BR1 instance is presented in figure 10.

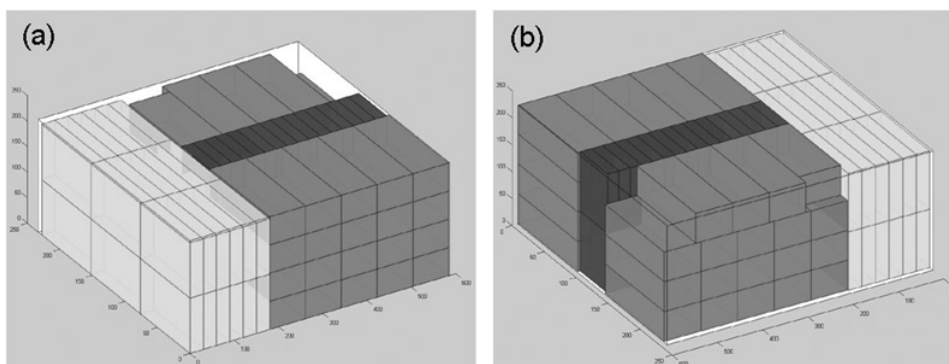


Fig. 10. A packing solution for the first instance from the BR1 set: (a) front view, and (b) back view.

With low timeout execution we achieved good results mainly for BR1 - BR3, the weakly heterogeneous problem instances. It is possible gain efficiency with set up a higher value to timeout execution and by implementing new heuristics to build layers across the container.

In order to prove the efficiency of our algorithm in to find good arrangements such way that a good weight distribution is achieved, we show the obtained results for Bischoff/Ratcliff test cases. We used the following formula to create boxes with different densities: $weight = (width / (2 * 10^5)) \times (w \times h \times d)$, where $w / (2 * 10^5)$, where w , h and d represent the width, height and depth of the box, respectively.

We considered the crossing-over parameter equals to 0.75, mutation parameter equal to 0.3, population size equal to 64 and maximum number of generations equals to 100. The table 2 presents the average length of chromosome (length), the less distance between center of mass of cargo and the geometric center of container (dist), the diagonal of container (diag), the average quality of solution (qwd) and its standard deviation (σ) and the execution time in seconds.

Group	length	dist.	diag.	qwd(%)	σ	time(s)
BR1	8.73	100.3	626.87	16	0.19	5.1
BR2	11.01	70.4	626.87	11	0.17	10.7
BR3	15.44	73.5	626.87	12	0.17	31.7
BR4	17.7	75.06	626.87	12	0.19	0.93
BR5	19.97	45.15	626.87	8	0.15	1.04

Table 2. Best-case results of Nepomuceno et al. (Nepomuceno et al., 2007) and the present methodology.

From the table 2 we affirm that chromosome length growing allows to diversify the arrangements of cargo, even and odd rotation discussed in section . However this characteristic needs improving, all weight distribution are relatively near to geometric center of container, using boxes with very different values of density.

The presented methodology achieved good results, mainly to weakly heterogeneous problem instances. However, to Bischoff/Ratcliff test cases, the weight distribution (quality of solution) is better to heterogeneous instances. The algorithm also reduces the common trade-off between an efficient use of container and the weight distribution of cargo.

In figure 11 we compare the results considering the weight distribution and use of container.

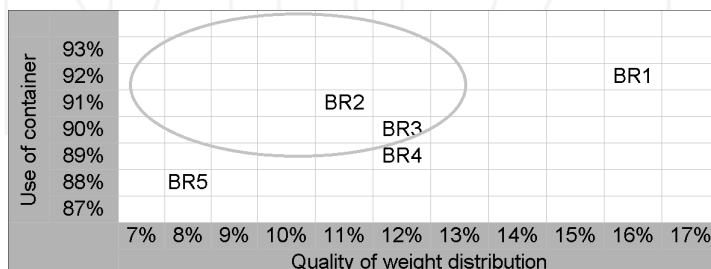


Fig. 11. Comparative of results using different benchmarks.

As we are interested in a maximum use of container and a coefficient of weight distribution (*qwd*) near to zero, the region we aims is the circle region in figure 11. Thus, the best achieved results were BR2 and BR3. This graphic aims help us to rank the results considering use of container and weight distribution.

5. Conclusion and future work

We introduced a hybrid methodology, the Heuristics Backtracking, an approach that combines a search algorithm, the backtracking, integer linear programming and genetic algorithms to solve the three dimensional knapsack loading problem considering weight distribution.

We discussed about the importance of weight distribution, among others generally unconsidered practical requirements, in practical situations in order to avoid damage of cargo during its transport by ships, trucks or airplanes. We also proposed a needed methodology to compare the quality of solutions 3.2.1, even to different benchmark tests or approaches' results.

Finally, we showed that the Heuristics Backtracking (HB) achieved good results without the commonly great trade-off between the utilization of container and a good weight distribution. Some benchmark tests taken from literature were used to validate the performance and efficiency of the HB methodology as well its applicability to cutting problems.

There are promising lines of investigation. Some of them to improve the already good results mainly regard with use of container and execution time, others to apply the HB to other kind of problems. So, we intend:

- To approach other variants of Container Loading Problem, for example the strip packing, placing boxes on shelves (Hoare & Beasley, 2001), load bearing ability (Ratcliff & Bischoff, 1998) among others;
- To implement heuristics those solve non-linear programming models to avoid many parameterized calls. It will improve the execution time of first phase;
- To improve the weight distribution by increasing the chromosome length. Some built blocks can be rotated in more than one way, so they can be manipulated by more than one gene. This changing will allow us to increase the quantity of discrete values for individuals' fitness;
- To change the backtracking algorithm, equivalent to a brute-force search. It intends to optimize the entire algorithm's time;
- An application of present methodology is in progress to solve cutting problems, similarly to what was made by other hybrid methodologies (Nepomuceno et al., 2008; Pinheiro et al., 2011).

6. Acknowledgements

This work has been financially supported by CNPq/Brazil via Grants #308249 / 2008-9, #473454 / 2008-4, and #312934 / 2009-2.

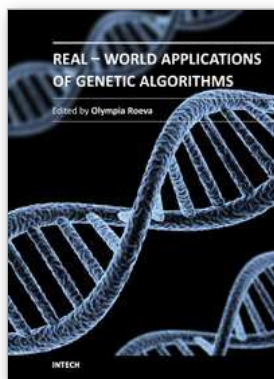
The authors also acknowledge IBM for making the IBM ILOG CPLEX Optimization Studio available to the academic community.

7. References

- Araújo, L.J.P. (2011). A Hybrid Methodology to solve the Container Loading Problem with Weight Distribution and Cutting Problems. *Master's thesis, Graduate Program in Applied Informatics at University of Fortaleza.*
- Araújo, L.J.P., Pinheiro, P.R. (2010). Combining Heuristics Backtracking and Genetic Algorithm to Solve the Container Loading Problem with Weight Distribution. *Advances in Intelligent and Soft Computing* 73:95-102.
- Araújo, L.J.P., Pinheiro, P.R. (2010). Heuristics Backtracking and a Typical Genetic Algorithm for the Container Loading Problem with Weight Distribution. *Communications in Computer and Information Science* 16:252-259.
- Araújo, L.J.P., Pinheiro, P.R. (2011). Applying Backtracking Heuristics for Constrained Two-dimensional Guillotine Cutting Problems. *Lecture Notes in Computer Science* 7030:113-120, to appear.
- Bischoff, E.E. (2006) Three dimensional packing of items with limited load bearing strength. *European Journal of Operational Research* 168:952-966.
- Bischoff, E.E., Ratcliff, M.S.W. (1995). Issues in the Development of Approaches to Container Loading. *Omega* 23:4:377-390.
- Bischoff, E.E., Janetz, F., Ratcliff, M.S.W. (1995). Loading Pallets with non-identical Items. *European Journal of Operational Research* 84:681-692.
- Bortfeldt, A., Gehring, H. (1998) Ein Tabu Search-Verfahren für Containerbeladeprobleme mit schwach heterogenem Kistenvorrat. *OR Spektrum* 20:237-250.
- Bortfeldt, A., Gehring, H. (2001) A Hybrid Genetic Algorithm for the Container Loading Problem. *European Journal of Operational Research* 131:143-161.
- Bortfeldt, A., Gehring, H., Mack, D. (2003). A Parallel Tabu Search Algorithm for Solving the Container Loading Problem. *Parallel Computing* 29:641-662.
- Chen, C.S., Lee, S.M., Shen, Q.S. (1993). An analytical model for the container loading problem. *European Journal of Operations Research* 80:6876.
- Christensen, S.G., Rousøe, D.M. (2009). Container loading with multi-drop constraints. *International Transactions in Operational Research* 16:6:727-743.
- Davies, A.P., Bischoff, E.E. (1999). Weight distribution considerations in container loading. *European Journal of Operations Research* 114:509-527.
- Derelia, T., Dasb, G.S. (2010). A hybrid simulated annealing algorithm for solving multi-objective container-loading problems. *Applied Artificial Intelligence* 24:5:463-486.
- Dumitrescu, I., Stuetzle, T. (2003). Combinations of local search and exact algorithms. *Applications of Evolutionary Computation LNCS* 2611:211-223.
- Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research* 44:145-159.
- Egeblad, J., Pisinger, D. (2009) Heuristic approaches for the two- and three-dimensional knapsack packing problem. *Computers & Operations Research* 36:1026-1049.
- Eley, M. (2002). Solving container loading problems by block arrangement. *European Journal of Operational Research* 141:393-409.
- Faroe, O., Pisinger, D., Zachariasen, M. (2003). Guided local search for the three-dimensional bin packing problem. *Inform Journal on Computing* 15:3:267-283.
- Gehring, H., Bortfeldt, A. (1997). A Genetic Algorithm for Solving the Container Loading Problem. *International Transactions in Operational Research* 4:401-418.
- Gehring, H., Bortfeldt, A. (2002). A Parallel Genetic Algorithm for Solving the Container Loading Problem. *International Transactions in Operational Research* 9:497-511.

- Gehring, H., Menschner, K., Meyer, M. (1990) A computer-based heuristic for packing pooled shipment containers. *European Journal of Operational Research* 44:277-288.
- George, J.A., Robinson, D.F. (1980). A heuristic for packing boxes into a container. *Computers and Operations Research* 7:147-156.
- Gilmore, P.C., Gomory, R.E. (1963). A linear programming approach to the cutting stock problem - Part II. *Operations Research* 11:863-888.
- Gilmore, P.C., Gomory, R.E. (1965). Multistage cutting and packing boxes into a container. *Computer and Operations Research* 13:94-120.
- Gilmore, P.C., Gomory, R.E. (1965). Multistage cutting stock problems of two and more dimensions. *Operations Research* 14:1045-1074.
- Glover, F., Laguna, M. (1993). Tabu search, In: C.R. Reeves (Ed.), *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications - Oxford, 70-150. (1993)
- Goldberg, D. E. (1989) Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley.
- Hoare, N.P., Beasley, J.E. (2001) Placing boxes on shelves: a case study. *Journal of the Operational Research Society* 52:6:605-614.
- Holland, J.H. (1975) Adaptation in Natural and Artificial Systems. *University of Michigan Press*.
- Holland, J.H. (1992) Adaptation in Natural and Artificial Systems, 2nd edition, *The MIT Press*.
- Kirkpatrick, S.; Gelatt Jr., C.D.; Vecchi, M.P.: Optimization by Simulated Annealing. *Science* 220, 4598:671-680 (1983)
- Levinson, M.: The box: how the shipping container made the world smaller and the world economy bigger. Princeton University Press. (2008)
- Liang, S., Lee C., Huang, S.: A Hybrid Meta-heuristic for the Container Loading Problem. *Communications of the IIMA*. 73:7:4 (2007)
- Loh, T.H., Nee, A.Y.C. (1992) A packing algorithm for hexahedral boxes. *Proceedings of the Conference of Industrial Automation*, 115-126.
- Mack, D., Bortfeldt, A., Gehring, H.: A parallel hybrid local search algorithm for the container loading problem. *International Transactions in Operations Research*, 11:511-533. (2004)
- Michalewicz, Z. (1992) Genetic Algorithms + Data Structures = Evolution Programs. *New York: Springer-Verlag*.
- Morabito, R., Arenales, M.: An and/or-graph approach to the container loading problem. *International Transactions in Operational Research*. 1:59-73 (1994)
- Murata, H., Fujiyoshi, K., Nakatake, S., Kajitani, Y.: VLSI module packing based on rectangle-packing by the sequence pair. *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*. 15:1518-1524. (1996)
- Nepomuceno, N., Pinheiro, P.R., Coelho, A.L.V.: Tackling the Container Loading Problem: A Hybrid Approach Based on Integer Linear Programming and Genetic Algorithms. In: VII European Conference on Evolutionary Computation in Combinatorial Optimization (EVOCOP). Berlin: Springer, 2007. v.4446. p.154 - 165. (2007)
- Nepomuceno, N.V., Pinheiro, P.R., Coelho, A.L.V. (2008) A Hybrid Optimization Framework for Cutting and Packing Problems: Case Study on Constrained 2D Non-guillotine Cutting. In: C. Cotta and J. van Hemert (Eds.), *Recent Advances in Evolutionary Computation for Combinatorial Optimization*, Chapter 6, pp. 87-99, Book of the Series "Studies in Computational Intelligence", Vol. 153, Springer-Verlag (ISBN: 978-3-540-70806-3).

- Ngoi, B.K.A., Tay, M.L., Chua, E.S. (1994) Applying spatial representation techniques to the container packing problem. *International Journal of Production Research* 32:111-123.
- Peng, Y., Zhang, D., Chin, F.Y.L.: A hybrid simulated annealing algorithm for container loading problem. Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, ISBN:978-1-60558-326-6, pp.919-928 (2009)
- Pinheiro, P.R., Coelho, A.L.V., Aguiar, A.B., Bonates, T.O. (2011) On the Concept of Density Control and its Application to a Hybrid Optimization Framework: Investigation into Cutting Problems. *Computers & Industrial Engineering* v.61:3:463-472.
- Pisinger, D. Heuristic for the Container Loading Problem. *European Journal of Operational Research*, 141:382-392. (2000)
- Pisinger, D.: Heuristics for the Container Loading Problem. *European Journal of Operational Research*. 141:143-153. (2002)
- Plateau, A., Tachat, D., Tolla, P.: A Hybrid Search Combining Interior Point Methods and Metaheuristics for 0-1 Programming. *Int. T. Oper. Res.* 9:731-746 (2002)
- Ratcliff, M.S.W.: Incorporating weight aspects into container loading approaches. EBMS Working Paper, University College, Swansea. (1994)
- Ratcliff, M.S.W., Bischoff, E.E. (1998) Allowing for weight considerations in container loading. *Operations Research Spektrum* 20:65-71.
- Wäscher, G., Haußner, H., Schumann, H.: An improved typology of cutting and packing problems. *European Journal of Operational Research*, Vol. 183, No.3. pp.1109-1130. (2007)
- Wolpert, D.H., Macready, W. G. (1997) No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions* 1:1:67-82



Real-World Applications of Genetic Algorithms

Edited by Dr. Olympia Roeva

ISBN 978-953-51-0146-8

Hard cover, 376 pages

Publisher InTech

Published online 07, March, 2012

Published in print edition March, 2012

The book addresses some of the most recent issues, with the theoretical and methodological aspects, of evolutionary multi-objective optimization problems and the various design challenges using different hybrid intelligent approaches. Multi-objective optimization has been available for about two decades, and its application in real-world problems is continuously increasing. Furthermore, many applications function more effectively using a hybrid systems approach. The book presents hybrid techniques based on Artificial Neural Network, Fuzzy Sets, Automata Theory, other metaheuristic or classical algorithms, etc. The book examines various examples of algorithms in different real-world application domains as graph growing problem, speech synthesis, traveling salesman problem, scheduling problems, antenna design, genes design, modeling of chemical and biochemical processes etc.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Luiz Jonatã Pires de Araújo and Plácido Rogério Pinheiro (2012). A Hybrid Methodology Approach for Container Loading Problem Using Genetic Algorithm to Maximize the Weight Distribution of Cargo, Real-World Applications of Genetic Algorithms, Dr. Olympia Roeva (Ed.), ISBN: 978-953-51-0146-8, InTech, Available from: <http://www.intechopen.com/books/real-world-applications-of-genetic-algorithms/a-hybrid-methodology-approach-for-container-loading-problem-using-genetic-algorithm-to-maximize-the->

INTeCH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821