A GRASP Approach to the Container Loading Problem

Ana Moura[a,c], José Fernando Oliveira[b,c]
[a]ESTIG - Escola Superior de Tecnologia e Gestão, Bragança, Portugal
[b]FEUP - Faculdade de Engenharia da Universidade do Porto, Portugal
[c]INESC-Porto – Instituto de Engenharia de Sistemas e Computadores, Porto, Portugal

Abstract
In order to solve a single Container Loading Problem a "wall building" constructive algorithm is presented. The performance of the constructive algorithm is improved by applying the GRASP meta-heuristic.
The performance of the GRASP approach is tested with test problems provided by (Bischoff and Ratcliff 1995). Results achieved for volume utilization and load stability are compared with well-known algorithms from the literature, both when homogeneous and heterogeneous types of cargo are considered.

1. Introduction

The container loading is a three dimensional problem that determines arrangements of items in a container. Usually the main objective of the container loading problem (CLP) is defined by the maximization of the efficiency of the loading space utilization. In this paper the problem involves only one container with known dimensions and the cargo varies from weakly to strongly heterogeneous. Only three constrains are considered: orientation constraint for the boxes, the cargo stability constraint and the container volume constraint.
The CLP is a special instance of the general class of problems referred to in the literature as cutting and packing. Using the (Dyckhoff 1990) classification, the container loading problem in our consideration can be denoted by: 3/V/I where the entire list of items is to be placed in to a single container. The main objective in such formulation is to maximize the volume of cargo accommodated which is equivalent to minimizing the waste space in the container. (George and Robinson 1980), (Morábito and Arenales 1993), (Bortfeldt and Gehring 1998), (Bortfeldt and Gehring 2001), (Bortfeldt, Gehring et al. 2002) and (Bischoff 2003) are examples of publications which deal with the single container loading problem. A number of approaches described in literature are tailored specifically to deal with efficiency of loading arrangement considerations. Some examples of this approaches which adopt a vertical wall building strategy are (George and Robinson 1980), (Bischoff and Marriot 1990), (Gehring, Menschner et al. 1990) and (Chien and Wu 1998). Heuristics procedures which build loading plans from a series of horizontal layers are also commonplace. Examples of solutions procedures that adopt a horizontal layering philosophy are (Bischoff and Ratcliff 1995) and (Ratcliff and Bischoff 1998). A further family of heuristics also build packing arrangements iteratively but not restrict configurations to consist of walls or layers. Often a single box is placed at each packing arrangement. Such publications include (Ngoi, Tay et al. 1994), (Davies and Bischoff 1999), (Eley 2002) and (Bischoff 2003).
Other approaches make use of more sophisticated operational research tools, like the meta-heuristics approaches. (Gehring and Bortfeldt 1997), (Bortfeldt and Gehring 1998), (Faina 2000), uses a tabu search and genetics algorithms in order to solve the container loading weakly or strongly heterogeneous problems. Later (Bortfeldt and Gehring 2001), (Gehring and Bortfeldt 2002), (Bortfeldt, Gehring et al. 2002) and

(Bortfeldt, Gehring et al. 2003) uses the parallelization of genetic and tabu search algorithms.
In section 2 the basic constructive heuristic based in (George and Robinson 1980) heuristic is described. Some modifications to the original (George and Robinson 1980) heuristic are made in order to improve his performance. In the following subsections those modifications are described. Following the GRASP paradigm an improved heuristic based in GRMod constrictive heuristic is discussed (section 3). In sections 4 and 5, results for the well known (Loh and Nee 1992) and (Bischoff and Ratcliff 1995) test problems are presented. First the results obtained with volume utilization objective function (section 4) are discussed. One of the goals rely on improve the load stability results. So some modifications in GRModGRASP are performed and reported in section 5. Section 6 summarizes the conclusion reached with this work. Another comparisons are carried out with other three approaches using the same basic "wall building" heuristic (GRMod) but different meta-heuristics. Those meta-heuristics are simulated annealing, tabu search and iterated local search. The results in terms of volume utilization, cargo stability and computational times are pointed out.

2. Changes applied to (George and Robinson 1980) heuristic

The new approach described in this paper is based on (George and Robinson 1980) heuristic. It is a "wall building" heuristic and the pack is performed along the depth direction. The container is open in the front and boxes are pack through this opening, from the back along his length.
One of the modifications to the (George and Robinson 1980) is related with the container. We consider a finite length to the container. With this modification we can eliminate the unsuccessful packing and the automatic repacking procedures of the packing algorithm. (George and Robinson 1980) heuristic was developed to deal with problems were the total volume of boxes is less than the total volume of the container. And those two procedures are to achieve a feasible solution in case of exist some unpacked boxes.
Another modification concerns the packing in the end of the container. The (George and Robinson 1980) heuristic uses a minimal length parameter that prohibit the packing of new layers in the end of the packing process. This cause sub approved layers. To avoid this in the GRMod heuristic the layer depth dimension was dependent of the volume of unpacked boxes. This way in the end of the container layers could have small depth but with better volume utilization.
Following the GRASP (Greedy Randomised Adaptative Search Procedure) paradigm the approach discussed in this paper is divided in two different steps. In the first step solution is built and in the second step this solution is improved with a local search algorithm. In the construction phase the container is loaded until one of the following three conditions is met: there are no more free spaces in the container; there are no more boxes to be packed; or the dimensions of the remaining free spaces are smaller than the dimensions of the boxes still available to pack. Afterwards a local search phase is run to improve this solution.

2.1. Constructive heuristic

Alike (George and Robinson 1980) heuristic, this constructive heuristic deals with empty spaces in two different ways. If an empty space has the same height and width than the container, then this space is treated by the heuristic as a new layer (section 2.1.1). In this case the layer's depth dimension is defined by the depth dimension of the

type of box chosen to start the layer. Otherwise the space is treated like a free space (section 2.1.2).
When a new layer is started the boxes are placed in vertical columns along the width of the container. In the other cases the algorithm tries to pack the boxes in the free spaces left by the boxes packed in the current layer and by previously built layers.
When the unpacked boxes do not fit in the free spaces then those spaces are temporarily marked as "rejected". The mark is only temporary because, if a new adjacent layer is built, this marked space can be amalgamated (Section 2.1.1.3) – as in the (George and Robinson 1980).

2.1.1. Building a layer

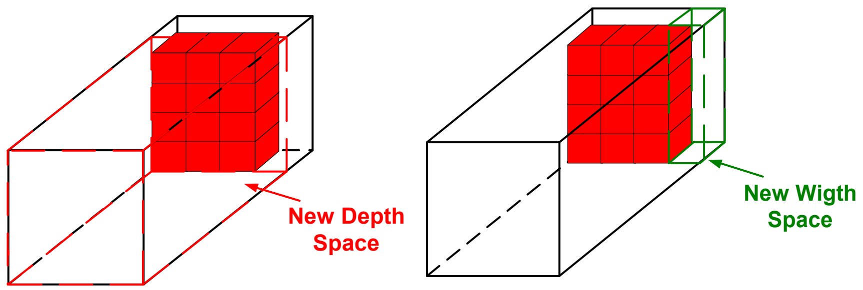
As previously referred, the heuristic is based in the "wall building" procedure. The container is filled with transversal walls and the depth of the layer is determined by the first box placed in the layer. In (George and Robinson 1980) heuristic the depth of the layers depends on a K parameter. This parameter is used limit the to depth dimension layer selection. One shortcoming in the original heuristic is the packing dependence of boxes ranking scheme. To eliminate this dependency, the box selecting ranking scheme and the box type status isn't used. All this are eliminated by performing a local search for the best box/orientation to open a new layer and to fill a space.

2.1.1.1. Starting a new layer

For all the boxes available this procedure computes the best arrangement when considering all possible orientations for each type of box. The best arrangement is found by simulating all the choices of boxes types and possible orientations and computing the correspondent volume utilization. If more then one arrangement yields the best volume utilization one of them is randomly chosen (Section 3) to become the definitive packing. After that the list of unpacked boxes and the list of free spaces are updated. The layer depth is equal to the box dimension placed along that direction. The number of boxes placed along the width and the height is limited by the container dimensions and the availability of that type of box. First the height of the container is filled as much as possible with an integer number of boxes and then these columns are replicated along the width of the container. An incomplete column is permitted.

2.1.1.2. New spaces generation

If there is some free space left between the layer and the container height or width, new spaces are generated (Figure 1) so that remaining boxes can be latter on packed there (Section 2.1.2).



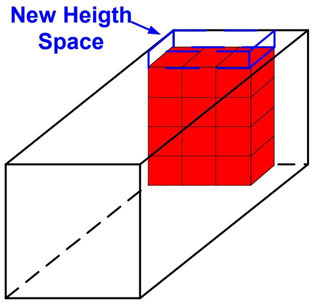


Figure 1 - Free spaces generation

Their generation follows a fixed order. The first space to be created is the depth space that corresponds to the frontal free space. This space is always created until the front of the container is reached. The next space to be generated is the width space and finally the height space is created. It should be noticed that, if the arrangement of boxes fits perfectly in the container along one of these dimensions, these spaces may have null dimensions, i.e. do not exist. In (George and Robinson 1980) heuristic at the time when the width space is created if the dimension is smaller than the minimum box dimension then the new width space is not accepted. Is this case the height space assumes the width of the original space (Figure 2). This results in no fully supported boxes. To improve the cargo stability, in this situation the GRMod heuristic assumes that: If one of the dimensions of the newly created spaces is smaller then the smallest dimension of the boxes not yet packed, then this particular space is marked as “rejected”. This way the approach guarantees that all boxes are fully supported.

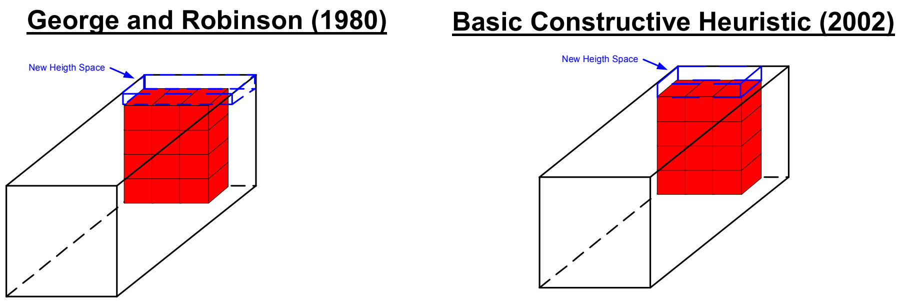


Figure 2 - Differences in space creation

All the generated spaces are placed in a list of spaces in the order by which they are generated. Later on, when free spaces are considered to pack boxes, they are used following a first-in-last-out strategy, favouring the full support of the packed boxes and increasing the cargo stability.

2.1.1.3. Amalgamation

When a new space is marked as “rejected” the algorithm tries to increase its size by amalgamating it with contiguous spaces belonging to the previous layer and also marked as “rejected”. By this procedure a new useful space may be generated (Figure 3). If no amalgamation can be performed with spaces of the previous layer then the “rejected” space is kept in the list hoping that, in the next layer, any new “rejected” space can be amalgamated with it.

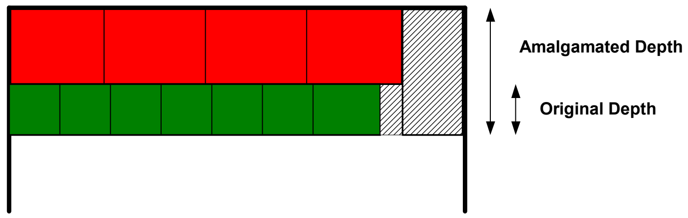


Figure 3 - Amalgamation procedure

It should be noticed that, by this process, an efficient and dense packing may be achieved. A direct consequence is that boxes with depth dimensions larger than the depth of the layer can now be packed there. This generates intersected walls.

2.1.1.4. Flexible Width

As stated before, very small spaces may be rejected as they have not been amalgamated with contiguous spaces and can not be used in the future to pack any boxes. To avoid this space fragmentation the original G&R heuristic proposed the concept of flexible width. This parameter bounds the number of columns that can be placed along the width in a new layer. Its value is propagated from the previous layer and is equal to the width of the arrangement of boxes that started the previous layer. For instance, if the previous layer was started with a box with a width of 30 cm and 4 columns were placed along the width, then the flexible width for the next layer would be 120 cm (Figure 4).
While (George and Robinson 1980) bound the number of columns in the new layer by taking the smallest integer that contains the flexible width, in the present algorithm the largest integer smallest than the flexible width will be taken. Taking the example presented in Figure 4, (George and Robinson 1980) heuristic would place an additional column in the new layer.

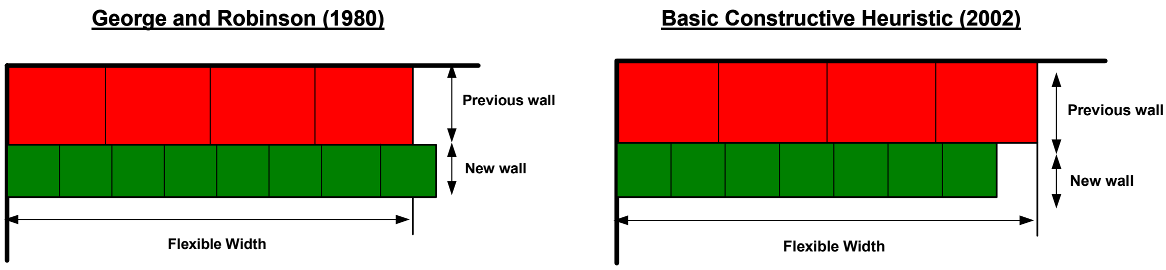


Figure 4 - Flexible Width for the New Layer

2.1.2. Filling a free space

The construction of a layer ends by filling the free spaces that were generated in the first step of the layer construction. The first space to be filled is the height space, following the previously mentioned last-in-first-out strategy (the height space was the last one to be created). Only the boxes that have smaller dimensions than the space dimensions are considered. For each type of box, the procedure computes all possible arrangements (number of columns in depth and width directions and number of boxes per column) and selects the one that yields the best volume utilization. Then for all best volume utilization arrangements one is randomly chosen, following the GRASP paradigm (Section 3), and the free space is filled with that box type and arrangement. When no feasible arrangement of boxes is found, this space is marked as “rejected”. Then the algorithm tries to amalgamate this space with any other previously marked spaces.
After filling a space new depth, width and height spaces are generated, processed and inserted in the spaces list. The last one to be inserted will be the first one to be used.

This filling spaces procedure is recursively applied until no more free spaces, different from the container front space, are available. In that case the new layer procedure is started applied to the container front space.

3. GRASP approach

Following the GRASP paradigm, randomization is used in this approach. In each iteration the choice of the next type of box to pack is made over a candidate list that contains the several alternatives of box types, ordered by volume utilization. A totally greedy strategy would lead to the choice of the best type of box (the first element of the candidate list) and a completely random strategy would draw from the entire list. However, a *restricted candidate list* (RCL) is built, with the best candidates. Then a random choice is made from this RCL list (Figure 5). The volume utilization tries to measure the benefit of selecting each type of box for a new layer or for a free space.
To define which candidates will belong to the RCL list a parameter α is used, which will control the level of greediness of the algorithm. This parameter can vary between [0,1]. After computing the volume utilization for all candidates (types of boxes) the RCL list is filled according to the following threshold:
$\beta = MVU + \alpha * (mVU - MVU)$ were:

- β is the volume utilization threshold;
- MVU is the maximum volume utilization computed for all possible arrangements;
- mVU is the minimum volume utilization computed for all possible arrangements;

If the volume utilization for one arrangement is bigger or equal to the β parameter, then the arrangement is added to the RCL. It is easy to see that when α=1, β is minimum and the basic heuristic is random; if α=0, β is maximum and the basic heuristic is greedy.

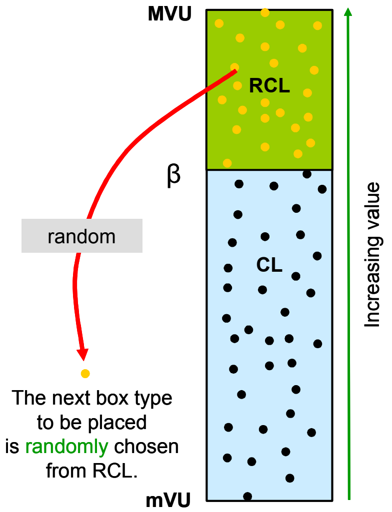


Figure 5 - Restricted candidate list

In the local search phase the algorithm starts with the solution built in the construction phase. Then a neighbourhood of this solution is built. If a better solution is founded in the neighbourhood, then it becomes the new current solution and a new neighbourhood is built around this new better solution. The local search procedure stops when no better solutions in the neighbourhood are found.

In order to build a neighbourhood a disturbance to the solutions must be defined (Figure 6). In this approach a position in the sequence by which the boxes were placed is randomly selected. Then all the boxes placed from that position until the end of the sequence are removed from the list of placed boxes and inserted in the list of unpacked boxes. The type of box that corresponds to the random position becomes "forbidden" and is temporary removed from both lists.

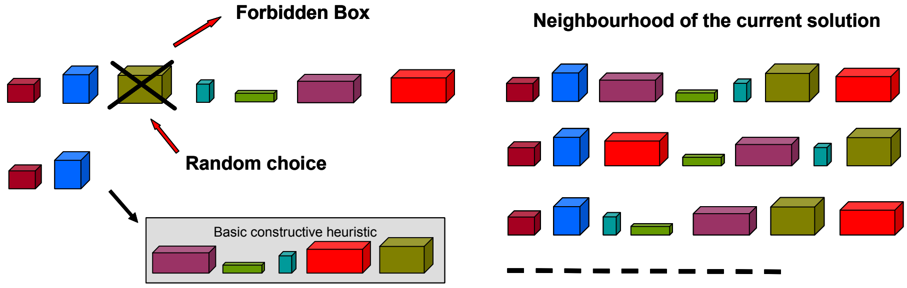


Figure 6 - Building a neighbourhood

Then, for all boxes belonging to the unpacked boxes list the heuristic applies the constructive heuristic, but now without any randomness (α=0). After the packing the first type of boxes the "forbidden" type of box is reinserted in the list of unpacked boxes. By this the box type that previously occupied the disturbed position will not retake that place. The constructive heuristic, in its greedy flavour, continues until no more boxes can be packed and a new solution is obtained.

4. Computational Tests

Standard test cases from literature were used for benchmarking purposes. (Loh and Nee 1992) generated 15 test cases named LN problems. Each test case used a different sized container. The container volume was large enough to pack all items in 13 of the 15 test cases. The number of different types of boxes lay between 6 to 10 and the number of available items varied between 100 and 250. The only constraint that is explicitly required is the orientation constraint.

Other test cases were compared, (Bischoff and Ratcliff 1995) and (Davies and Bischoff 1999) generated 15 classes BR1 to BR15 with 100 test cases each. The classes vary from weakly to strongly heterogeneous problems. Test cases in BR1 use three different types of items and this number is increased to hundred for the test cases in BR15. According to the decreasing average number of box per box type in the test case BR1 there are on average 50.2 boxes for each box type, but in test case BR15 the average number is only 1.30. For all types the length of the item's edges were integers numbers chosen from intervals [30,120], [25,100] and [20,80] respectively. A standard ISO container is used. Again only the orientation constraint is to be met.

For benchmarking the GRModGRASP heuristic performance was compared with the following nine approaches:

- H_B_al.: the heuristic approach of (Bischoff, Janetz et al. 1995);
- H_BR: the heuristic approach of (Bischoff and Ratcliff 1995);
- GA_GB: the genetic algorithm of (Gehring and Bortfeldt 1997);
- TS_BG: the tabu search approach of (Bortfeldt and Gehring 1998);
- HGA_BG: the hybrid genetic algorithm of (Bortfeldt and Gehring 2001);
- PGA_GB: the parallel genetic algorithm of (Gehring and Bortfeldt 2002);
- H_E: the heuristic approach of (Eley 2002);
- PTS_B_al.: the parallel tabu search approach of (Bortfeldt, Gehring et al. 2003)
- H_B: the heuristic approach of (Bischoff 2003);

The first performance evaluation criterion of a container loading algorithm is the volume utilization. Table 1 shows the results for the 15 LN test cases. When a method can achieved the best known volume utilization for a problem instance is stated that a “Best Value” is achieved. And when a method has packed all boxes of a problem instances in the container is stated that a “Global Optima Solution” is achieved. As long as we know, only seven of the nine comparing approaches had published results of these test cases. Related to mean value of volume utilization the best results are achieved by the TS_BG and PTS_B_al. approaches. Those approaches can achieved fifteen best values, two more than the others. Looking to the mean value of volume utilization the GRModGRASP outperforms the remaining approaches and achieves the same number of global optima solutions than the other methods.

	H_B_al.	H_BR	GA_GB	TS_BG	HGA_BG	H_E	PTS_B_al.	GRModGRASP
Mean Value of Vol. Util. (%)	69,5	68,6	70,0	70,9	70,1	69,9	70,9	70,3
Best values	10	11	12	15	13	13	15	13
Global Optima Solution	10	11	12	13	13	13	13	13

Table 1 - Results for the LN test classes

In Figure 7 are presented the results for the 15 BR problems. Some approaches only report the results for weakly heterogeneous problems (from BR1 to BR7). Does approaches are H_E, H_B and PTS_B_al.. The average volume utilization for all approaches decreases with the increasing of cargo heterogeneity. But some approaches have greater bending curve than others. Comparing the variation of volume utilization for all BR problems we can stat that the GRModGRASP approach competed well with the other ones.

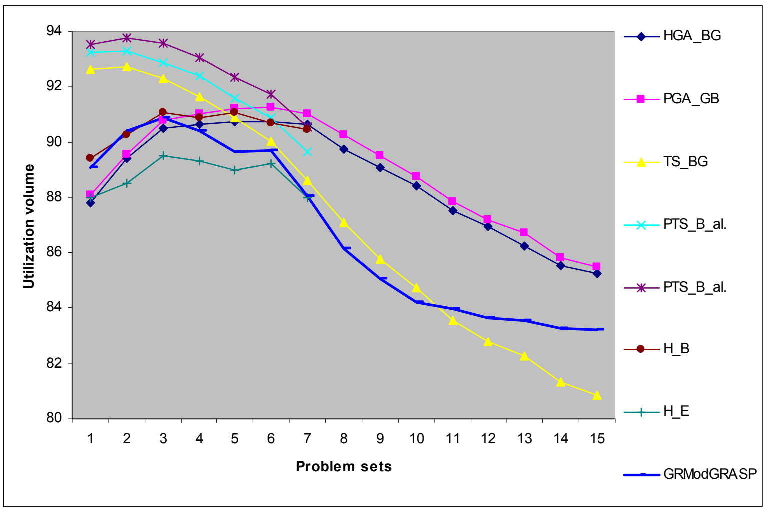


Figure 7 - Results for the BR test cases

In order to simplify the results comparison the 15 BR problem sets are divided in two groups: the weakly heterogeneous (from BR1 to BR7) and strongly heterogeneous (from BR8 to BR15).

Comparing weakly heterogeneous problems results, only three approaches always achieved better results than GRModGRASP, does are the two PTS_B_al. approaches and TS_BG approach. But for strongly heterogeneous problems the TS_BG is outperformed by GRModGRASP for the last five BR problem sets. For the first three

problems the H_B and the GRModGRASP achieved very close results and H_E is outperformed by the GRModGRASP for all problem sets.
For the first three BR problem sets, the PGA_GB and HGA_BG have worse volume utilization than GRModGRASP. But with the increasing of the cargo heterogeneity they achieved better results than GRModGRASP. In this kind of problems (strongly heterogeneous problems) the genetics algorithms PGA_GB and HGA_BG always achieve better results.

5. Cargo stability analysis

To guarantee that the cargo is not damaged during the transportation, the stability is one of the must important aspects to consider in the container loading problem. A cargo is stable if two situations occur:

- All boxes are fully supported;
- All boxes has at least three sides supported;

(Bischoff and Ratcliff 1995) presents two measurements to evaluate the cargo stability. The first one named Measurement1 gives the average number of boxes by which items other then those on the container floor are supported. This measurement is not used in our approach because the algorithm guaranties that all boxes are fully supported. The second measurement - measurement2, gives the average percentage of boxes not surrounded on at least three sides. So as smaller it is as better it is. We evaluate the packing produced by the GRModGRASP approach and compare the results with the results available in the literature.
The Measurement2 results available in the literature are only for the first seven BR problems. And this comparison is shown in the Figure 8. It can be seen that H_B_al. olds the best results but GRModGRASP results are very competitive.

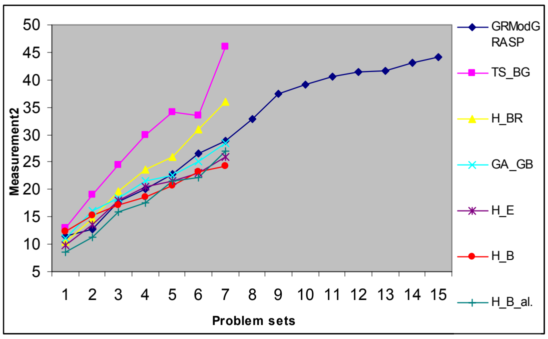


Figure 8 - Stability results for BR problem instances

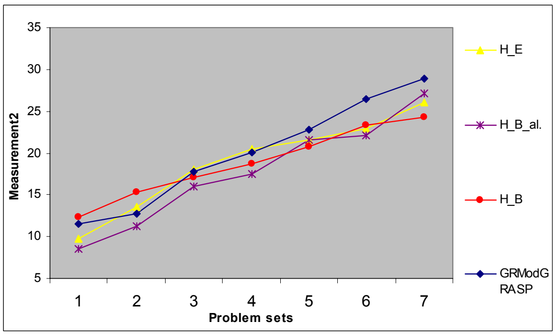


Figure 9 – Comparing the best stability results for BR problem instances

In order to improve the cargo stability, we try to increase the number of supported sides of each box. Two modifications in the constructive heuristic are made:

1. Reverse the walls in the container;
2. Change the objective function of the problem;

The first modification arises because some walls are higher than others. Looking to Figure 10, the small boxes placed over the second wall (wall 2) only has one or two supported sides if the box is in one end or in the middle of the set, respectively. If this wall is reversed those boxes became with two or three supported sides. This way the measurement2 could decrease.

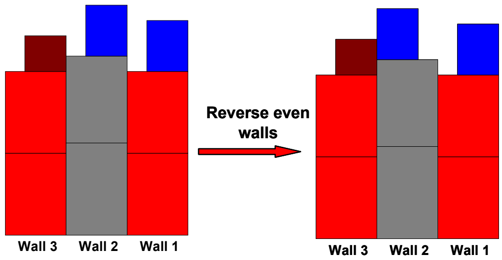


Figure 10 - Reverse the walls

The other modification is related with the objective function. In the constructive heuristic we change the objective function to: Maximize the number of supported boxes sides. To fill a space or to open a wall, the choice of the next box type is selected as usual from the RCL. But now the candidates belonging to the RCL are chosen by: the best arrangements that maximize the number of supported boxes sides. In case of tie the maximum volume utilization of the arrangement is used.

5.1. Results of the experiments

Tests are made for the weakly heterogeneous BR problems. As the expected, with the reverse walls approach, volume utilization results stay the same. Likewise, cargo stability results are practically the same. This is owing to some particular situations. For example in Figure 11 with the original heuristic one of the little boxes placed in the end of the container, has three sides supported. Reversing the walls, the same box becomes only with two sides supported. In these sorts of cases, the Measurement2 increases when compared with the original GRModGRASP. So the mean value of measurement2 is worse. And for weakly heterogeneous problems the cargo stability average stays practically the same. For strongly heterogeneous problems does not make sense use this modification in the original approach. Because in a strongly heterogeneous pack the arrangements of boxes are not walls, but only columns of different types of boxes (Figure 12).

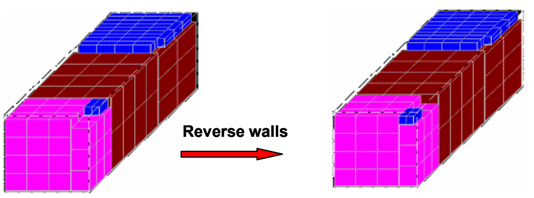


Figure 11 - Reverse walls for weakly heterogeneous problems

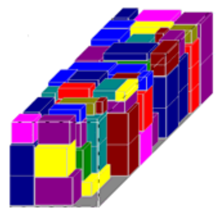


Figure 12 - Reverse walls for strongly heterogeneous problems

Related to the second proposed modification, the change of objective function, some tests are made to the weakly and strongly heterogeneous BR problems. For all the BR problem sets the same conclusions is taken. The cargo stability stays practically the same but the volume utilization decrease significantly. None of these two modifications in the GRModGRASP approach are taken in account.

6. Conclusions

Taken in to consideration the performance of GRModGRASP approach we can conclude that in terms of volume utilization some approaches perform better for weakly heterogeneous problems and others perform better for strongly heterogeneous problems. Considering not only the volume utilization, but the cargo stability and the computation time we can stat that GRModGRASP has a good performance. This approach proved that is an efficient method to apply to both weakly and strongly heterogeneous problems. The running times are very small even to problems with 100 different types of boxes (Figure 13).

Other approaches are tested with the same basic constructive heuristic GRMod. The meta-heuristics applied are: Simulated annealing (GRModSA); Tabu Search (GRModTS); and Iterated Local Search (GRModILS).

Comparing these three approaches with the GRModGRASP, as we can see in Figure 13 the volume utilization achieved with GRModGRASP for weakly heterogeneous problems are higher than with the others approaches. For strongly heterogeneous problems the GRModSA can almost equal the results achieved with the GRModGRASP.

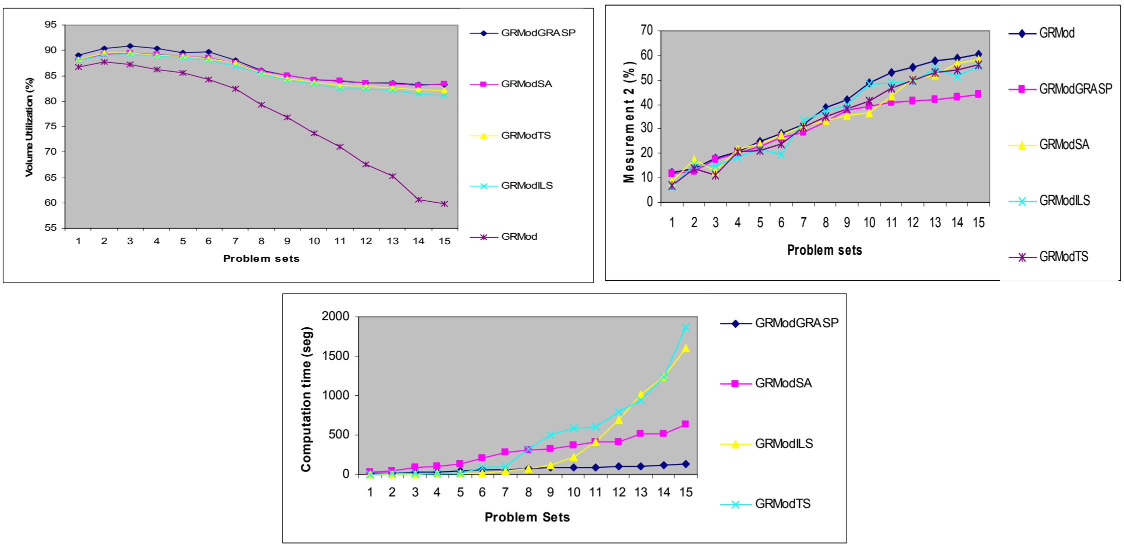


Figure 13 - Volume utilization, cargo stability and running times results for the four meta-heuristics

As we can see in Figure 13, for strongly heterogeneous problems cargo stability is better applying the GRASP meta-heuristic.

The running times for all approaches are really insignificant for weakly heterogeneous problems. But when the number of different types of boxes increases the GRModGRASP approach outperforms the other ones. And the running time is smaller than 200 seconds even to 100 different types of boxes.

Bibliography

Bischoff, E. E. (2003). Dealing with load bearing strength considerations in container loading problems. Swansea, European Business Management School University of Wales Swansea**:** 17.

Bischoff, E. E., F. Janetz, et al. (1995). "Loading pallets with non-identical items." European Journal of Operational Research **84**: 681-692.

Bischoff, E. E. and M. D. Marriot (1990). "A comparative evaluation of heuristics for container loading." European Journal of Operational Research **44**: 267-276.

Bischoff, E. E. and M. S. W. Ratcliff (1995). "Issues in the development of approaches to container loading." Omega, International Journal of Management Science **23**: 377-390.

Bortfeldt, A. and H. Gehring (1998). "A tabu search algorithm for weakly heterogeneous container loading problems." OR Spektrum **20**: 237-250.

Bortfeldt, A. and H. Gehring (2001). "A hybrid genetic algorithm for the container loading problem." European Journal of Operational Research **131**: 143-161.

Bortfeldt, A., H. Gehring, et al. (2002). A parallel tabu search algorithm for solving the container loading problem. D. N. 324, Diskussionsbeitrage des Fachbereichs Wirtschaftswissenschaft der FerUniversitat Hagen.

Bortfeldt, A., H. Gehring, et al. (2003). "A parallel tabu search algorithm for solving the container loading problem." Parallel Computing **29**: 641-662.

Chien, C. F. and W. T. Wu (1998). "A recursive computational procedure for container loading." Computers and Industrial Engineering **35**: 319-322.

Davies, A. P. and E. E. Bischoff (1999). "Weight distribution considerations in container loading." European Journal of Operational Research **114**: 509-527.

Dyckhoff, H. (1990). "A typology of cutting and packing problems." European Journal of Operational Research **44**: 145-159.

Eley, M. (2002). "Solving container loading problems by block arrangement." European Journal of Operational Research **141**: 393-409.

Faina, L. (2000). "A global optimization algorithm for the three-dimensional packing problem." European Journal of Operational Research **126**: 340-354.

Gehring, H. and A. Bortfeldt (1997). "A genetic algorithm for solving the container loading problem." International Transactions in Operational Research **4**: 401-418.

Gehring, H. and A. Bortfeldt (2002). "A parallel genetic algorithm for solving the container loading problem." International Transactions in Operational Research **9**: 497-511.

Gehring, H., K. Menschner, et al. (1990). "A computer-based heuristic for packing pooled shipment containers." European Journal of Operational Research **44**: 277-288.

George, J. A. and D. F. Robinson (1980). "A heuristic for packing boxes into a container." Computers and Operational Research **7**: 147-156.

Loh, T. H. and A. Y. C. Nee (1992). A packing algorithm for hexahedral boxes. Proceedins of the Conference of Industrial Automation, Singapore.

Morábito, R. and M. N. Arenales (1993). An and/or graph approach to the container loading problem. São Paulo - Brasil, Universidade de São Paulo - ICMSC**:** 29.

Ngoi, B. K. A., M. L. Tay, et al. (1994). "Applying the spatial representation technique to the container packing problem." International Journal of Production Research **32**: 111-123.

Ratcliff, M. S. W. and E. E. Bischoff (1998). "Allowing for weight considerations in container loading." OR Spektrum **20**: 65-71.