THE INDUSTRIAL PLANTS RELOCATION:ISSUES, POLICIES, PROCEDURES AND ALGORITHMS FOR DISASSEMBLY AND REASSEMBLY PHASES

De Falco Massimo[(a)], Gaita Pellegrino[(b)], Ilsami Mario[(c))], Santillo Libertina Carmela[(d)],

[(a)] DIIN - Department of Industrial Engineering - University of Salerno- Italy
[(b),(c),(d)] D.I.C.Ma.P.I. - Department of Chemical, Materials and Industrial Production Engineering, University of Naples Federico II, P.le Tecchio, 80 – 80125 - Naples - Italy

[(a)] mdefalco@unisa.it, [(b)]pellegrino.gaita@unina.it, [(c)] mario.ilsami@unina.it , [(c)] santillo@unina.it,

ABSTRACT

This work proposes an analytical approach to optimize the transfer project of existing industrial plants in new productive locations. This means to transport in new sites the same functional characteristics that plants have in the original one. The impracticality to move existing plants, shifts the problem into define a transferring project, regarding constraints and opportunities related the execution of two strongly specular phases.

The aim is an algorithmic procedure which, selecting the main components of a plant, determines and optimizes the transfer global cost, as function of the parameters characterizing the three different project phases; disassembling, transporting and reassembling.

The core is constituted by an iterative structure which, starting from an analytical description plant's , integrates the loading and transport phases, estimating at the same time the best level of disassembly sequences and the loading efficiency and plant's reduction. Those evaluations allow to optimize the overall cost of the project.

Keywords: Algorithmic Modeling, Manufacturing, Logistics, Reassembly.

1. INTRODUCTION

Offshoring is the relocation by a company of a business process from one country to another.

Relocation of production, whereby a firm transfers all or part of its production facilities from one country to another, has become an important issue in industrial relations across Europe and especially in those western European countries that have seen a number of high-profile cases of outward relocation over recent years.

China and India have emerged as a prominent destinations for production offshoring.

The main reasons why companies resort to offshoring are:

- To increase efficiency;
- Because of the increased competition at the national level;
- to enhance image and prestige of the company.

The main benefits of offshoring are:

- Low costs and taxes;
- Time savings;
- The entry into new market areas.

However there are also risks related to the transfer of production abroad. First and foremost is the reduction in the level of employment and the loss of control and image quality.

In addition, others factors to consider are:

- Political climate in foreign countries;
- Differences in work practices and cultural barriers;
- Hidden costs.

The problem of plant transfer emerges when an already installed and operating plant has to undergo a change of location. The project of plant transfer can be divided in three sequential and temporally distinct phases:

1. Plant disassembly;
2. Transportation to the new production site;
3. Re-assembly.

The small components, obtained from disassembly, are loaded into containers and then they are transported by appropriate means of transport to the new site. Instead the singular transport is provided for large components. In particular small components determine significant problems that will be treated in this paper with a mathematic and algorithmic approach.

2. STATE OF ART FOR DISASSEMBLY

All Disassembly is defined as the process of separating products into parts and/or subassemblies with necessary inspections and sorting operations. Generally, disassembly is done with several purposes, i.e., recovering material fractions, isolating hazardous substances, separating reusable parts and/or subassemblies, etc. Note that disassembly is one of core operations in recovering or even disposing of used/end-of-life products.

Disassembly objectives include:

- Recovery of valuable parts or subassemblies,
- Retrieval of parts or subassemblies of discontinued products,

- Removal of hazardous parts,
- Increasing the purity of the remainder of the product,
- Achieving environmentally friendly manufacturing standards.

A first model has been formulated by Gupta and Gungor. It presents a methodology to obtain a near optimal DSP. A DSP is a sequence of disassembly tasks that begins with a product to be disassembled and terminates in a state where all the parts of the product are disconnected. The number of alternative DSPs increases exponentially with the number of parts in the product being disassembled. For example, a product with 10 parts may have up to a whooping 10! = 3,328,800 disassembly sequences. Even though the number of feasibility constraints may reduce the number of disassembly sequences, this number is still expected to be large for a complex product.

A second model has been formulated by three Korean engineers, named Han, Yu and Lee.

It considers the selective disassembly sequencing problem under the serial disassembly environment in which only one component is obtained at each disassembly operation. The problem is represented as a disassembly precedence graph and then a new integer programming model is suggested for the objective of minimizing the total disassembly cost. To solve the problem, we suggest a branch and fathoming heuristic.

The disassembly precedence graph, suggested by Lambert (2006), is adopted to represent disassembly operations and their precedence relations. More formally, the disassembly precedence graph can be represented by G = (N, A), where N\{0} is the set of nodes that represent disassembly operations, i.e., N = {0, 1, 2, … n} and A is the set of arcs that represent the precedence relations between two disassembly operations, i.e., A = {(i, j)| i, j ∈ N and i < j}. Also, node 0 denotes the root node representing the start of disassembly. Note that each disassembly operation is associated with a part or subassembly to be obtained from that operation since we consider the serial disassembly and the number in each node represents the corresponding part or subassembly.

The Container Loading problem (CLP) alludes to the task of packing boxes into containers. More precisely, given the dimensions of the containers and the boxes which need to be loaded, the problem can be defined as to find such an arrangement of boxes that optimizes a given objective function that, in general, is the maximum volume of the loaded boxes. In addition to the geometric constraints, other restrictions can also be considered, such as boxes orientation and cargo stability.

The container loading problem is to pack boxes into containers so as to meet certain objectives. Examples of objectives are:

(a) Minimize the length of the container required for a specified cargo.

(b) Maximize the volume of cargo packed into a container.

(c) Minimize the number of containers needed to pack a specified cargo.

(d) Find a way to pack all given cargo into a container.

2.1. George and Robinson's heuristic

J. A. George and D. F. Robinson (1980) present a heuristic for packing boxes into a container. It fills the container layer by layer across its width. A new layer is not commenced until all the previous layers are packed. Criterions are set to determine which box should be used to commence the next layer. When a box type is determined to commence the next layer, as many complete columns of boxes of this box type are filled into this layer. The unfilled space of this layer is cut into cubic spaces that are packed by boxes of other types. Then it will try to amalgamate gaps in this layer with the previous layers to see if it is possible to fill more boxes. One may re-order the priority of choosing the box type to reload the container to see if it is possible to fill the cargo specified.

2.2. The new heuristic

The new approach described in this paper is based on George and Robinson's 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's heuristic 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. Another modification concerns the packing in the end of the container. The George and Robinson's 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. Alike George and Robinson's 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. 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. 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, as in the George and Robinson's heuristic. The new heuristic is based on 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. A local search for the best box/orientation is performed to open a new layer and to fill a space. 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 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. 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.

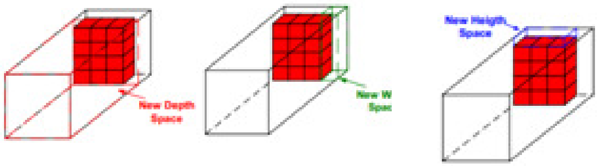
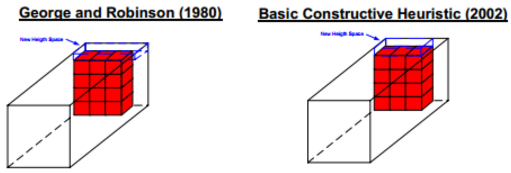


Figure 1. Example of space generation

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's 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).

Figure 2. Width of the original space

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 than 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.

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.

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.

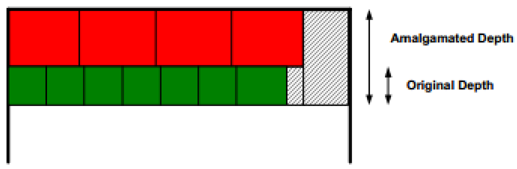


Figure 3. New useful spaces

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.

As stated before, very small spaces may be rejected as they have not been amalgamated with contiguous spaces and cannot 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's heuristic bounds 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's heuristic would place an additional column in the new layer.

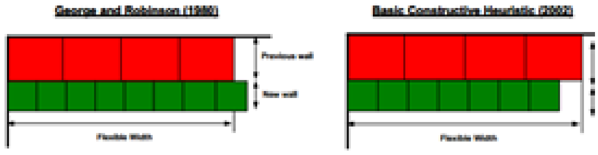


Figure 4. Flexible width for next layer

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, 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.

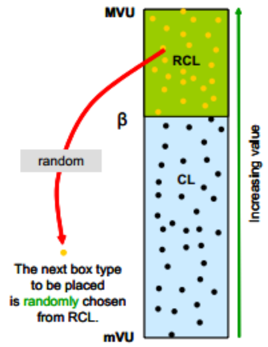


Figure 5. RCL list

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:

$\beta = MVU + \alpha * (mVU - MVU)$

where:

- 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.

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

In order to build a neighborhood a disturbance to the solutions must be defined (figure 6).

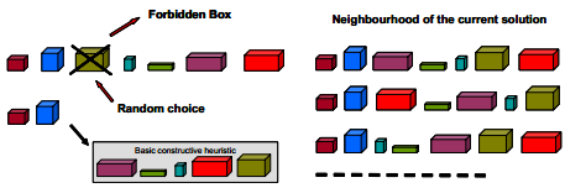


Figure 6. Example of defining disturbance

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. 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 flavor, continues until no more boxes can be packed and a new solution is obtained.

3. REASSEMBLY

Reassembly is the last phase of the transport process of an industrial plant.

All major components, which were previously disassembled and then transported to the new site, are cleaned to remove any residue of paint, rust, dirt, grease, oils, deposits of coal, coke and other contaminants. Each component is thoroughly inspected and the company determines whether it can be recovered and reused, if it must be repaired to go back to proper working condition or if it is sufficiently worn out or obsolete and it must be replaced with a new component. All recovered, refurbished and new core components are finally reassembled to create a remanufactured product whose performance is equal to that of a new product. A remanufactured product is expected to match current factory specifications, meet or exceed the original expected lifetime, and may even need to comply with the latest engineering specifications and environmental requirements.

3.1. Two types of reassembly

It is possible to distinguish two different types of reassembly:

- Symmetric reassembly;
- Asymmetrical reassembly.

In general, if our industrial plant is newly built and the aim is simply to transfer it to a new location so the reassembly phase is, in many instances, perfectly symmetrical with the disassembly phase. In other words, the reassembly sequence is exactly the reverse of the disassembly sequence. Instead, in the case of the transport of older generation systems, it is usually necessary to replace obsolete machines and components and to adopt through engineering studies a new and better layout configurations. In this case, therefore, the reassembly sequence may be asymmetrical with the disassembly sequence. The asymmetric reassembly is adopted for the following objectives:

- To optimize the space available;
- To speed up the transfer of material and/or persons from the various departments of the plant;
- To ensure, as far as possible, the proximity between departments where there is an important relationship.

4. THE PROPOSED MODEL

The proposed approach aims to overcome the traditional logic scheme, characterized by two sequential and independent phases: the separated determination of the disassembly sequence and the items loading into containers (Figure 7). A more deepened analysis would evidence that the cost connected the these two phases is function of the chosen level of plant disassembly, and that this will impact the disassembly and local re-assembly costs. Thus is evident that they would be lower if the plant remains assembled during the transport. The transport cost, on the contrary, would decrease with the raise of the disassembly level; it is widely known in literature that a reduction of the average dimensions of the items produces a more effective loading, with possible optimization of the number of containers and then transportation cost. These considerations suggest the possibility to find an optimum disassembly level which minimizes the sum of disassembly, transport and re-assembly costs. Therefore, a new logical scheme for the problem approaching is configured as shown in Figure 1-b. The phases of disassembly and loading are no longer independent: a further connection "area" will take into account all the possible disassembly modalities and evaluates them on the overall generated costs basis.

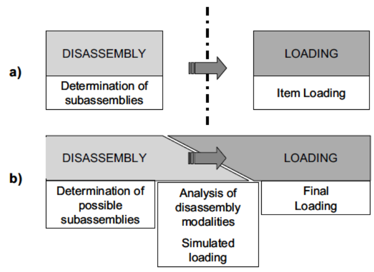


Figure 7. Determination of sequence

The structured procedure proposed for determining the optimum modality of plant transfer, according to the total cost minimization criterion, requires the following activities:

- analysis of the items subject to the procedure;
- disassembly/assembly sequences identification;
- identification of the optimum level and global costs calculation;
- determination of the Shipping List.

In Figure 2 the general flow chart of the procedure is illustrated. It is easy to notice the presence of an iterative block justified by the mutual dependence of the disassembly and loading/transportation phases.

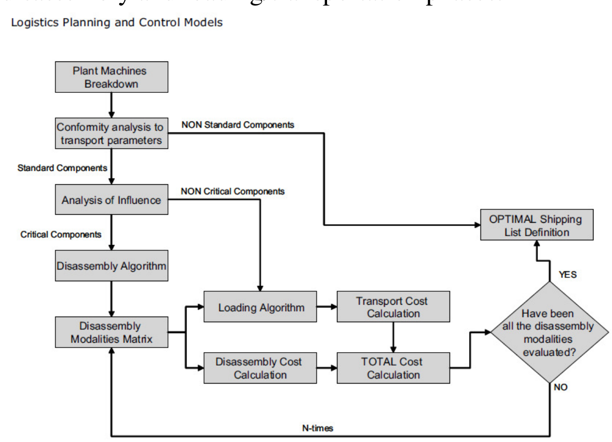


Figure 8. Logistics planning and control models

4.1. The algorithm

Given the dependence of the transport costs from the disassembly phase, the algorithm starts determining all the possible disassembly modalities, loading modalities and classifies them according to the resultant global cost.

In practice, the algorithm produces a M(NXR) matrix, in which N is the number of possible disassembly modalities and R is the number of disassembly actions present in one modality at least. Will be:

- $M_{ij} = 1$ if the jth action occurs in the with disassembly modality;
- $M_{ij} = 0$ in the contrary case.

		Actions							
		1	2	...	j	j+1	j+2	...	R
Modalities	1	0	0	0	0	0	0	0	0
	...								
	i	1	1	0	1	0	0	0	0
	i+1								
	...								
	N								

Figure 9. Matrix of disassembly modalities

The first line corresponds to the fully-assembled component and therefore is characterized by the 0 value for every action. Once obtained the matrix, the procedure goes through two paths:

- calculation of the disassembly/re-assembly cost;
- implementation of the container loading algorithm and subsequent calculation of the transport cost.

The disassembly/re-assembly cost relative to the with modality $C_{dis(i)}$ can be expressed as:

$$C_{DIS(i)} = (2\cdot C_{man}\cdot T_{OP(i)} + C_{ext}\cdot T_{ext})\cdot N_{wor} + C_{eq}$$

in which:

- C_{man} = manpower hourly cost [€/h];
- $T_{OP(i)} = \sum_{j=1}^{R} M_{ij} \cdot T_j$ = operation time associated with the ith disassembly modality [h]; the factor 2 means that the same time is assumed necessary also for the reassembly phase;
- T_j = time necessary for the jth action [h];
- N_{wor} = number of necessary workers;
- C_{ext} = travelling daily allowance [€/day];
- T_{ext} = number of necessary transfer days.
- C_{eq} = equipment cost.

The implementation of the container loading algorithm requires, for each disassembly modality, the list of components with relative dimensions. The result is the total number of containers necessary to contain the entire disassembled plant, which influences directly the transport cost.

At this stage will be possible to identify the disassembly/assembly level-sequence corresponding to the minimum total cost and, therefore, to define the optimal shipping list.

5. CONCLUSIONS

This thesis proposes a structured approach to industrial plant transfer problem; the main result is a analytical procedure which allows to correlate all possible disassembling-loading modalities of the standard-critical components with the relative global transfer cost, identifying, then, the optimum. The procedure consists of an iterative cycle in which, for each disassembly modality, a run of the container loading algorithm is performed. The sum of the consequent disassembly and transport costs determines the best disassembly sequence for the industrial plant.

Future developments of the research will be targeted to an increase of the procedure efficiency. In fact the number N of the disassembly sequences, being the industrial plant a complex object, can be extremely large; this means that the algorithm of container loading has to be run N times with the risk of not acceptable computational times. An interesting starting point could be the definition of a method to pre-select the "more promising" disassembly sequences in order to reach the best solution in a limited number of interactions. The risk of achieving only a local optimum solution should be compared with the advantages of the computational time saving.

REFERENCES

Converso, G., De Vito, L., Santillo, L., 2013. A simulation approach for sustainability of global supply chain network, 25th European Modeling and Simulation Symposium, EMSS

Converso, G., Gallo, M., Aveta, P., Santillo, L.C. Planning of supply chain risks in a make-to-stock context through a system dynamics approach, 2012

De Falco M., S. Miranda, 2007. "Analytical model for the optimization of industrial plant transfer projects", Proceedings of the 12th International Symposium on Logistics, 8-10 July 2007, Budapest.

Hee-Jong Han, Jae-Min Yu, Dong-Ho Lee, 2012. "Selective Disassembly Sequencing with Sequence-dependent Setups for Multiple Target Components", Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference 2012.

Gallo, M., Murino, T., Pensato, V., Santillo, L.C., 2013. A proposal for an exact algorithm for solving a SALBP-1 problem, Proceedings - 19th ISSAT International Conference on Reliability and Quality in Design, RQD 2013, pp. 383-388.

Gungor A., Gupta S.M., 1998. "Disassembly sequence planning for complete disassembly in product recovery", Proceedings of the 1998 Northeast Decision Sciences Institute Conference, Boston, Massachusetts, March 25-27.

Longo, F., Nicoletti, L.,Chiurco, A., Document Cooperative training for ships and tugboats pilots based on interoperable simulation 2013 25th European Modeling and Simulation Symposium, EMSS 2013 pp. 695-703

Moura A., José Fernando Oliveira, 2005. "A GRASP Approach to the Container Loading Problem", IEEE Computer Society.