

A WEB3D APPLICATION FOR THE BIN-PACKING PROBLEM

Bruno Fabio^(a), Caruso Francesco^(b), Pisacane Ornella^(c),
Musmanno Roberto^(d), Muzzupappa Maurizio^(e), Rende Francesco^(f), Giovanni Venuto^(g)

^{(a) (b) (c)}Università della Calabria – Department of Mechanical Engineering – Rende (CS) - Italy

^{(c) (d) (f) (g)}Università della Calabria – Department of Electronics, Informatics and Systems – Rende (CS) - Italy

^(a)f.bruno@unical.it, ^(b)francesco.caruso@unical.it, ^(c)opisacane@deis.unical.it,
^(d)musmanno@unical.it, ^(e)muzzupappa@unical.it, ^(f)frende@itkey.it

ABSTRACT

This paper describes a web application for supporting the user in the assessment of the optimal loading configuration for several carriers (i.e. trucks, containers, ship, etc.). The application has a standard form-based user-interface to insert data of the available carriers and items have to be loaded. The solution of this problem (known as Bin-Packing Problem (BPP)) is found by a specific algorithm and visualized by a 3D graphics representation inside the web page. Although some commercial applications already exist, ours runs directly on the web and offers, at the same time, an efficient and robust solver and a 3D visualization allowing the user to better understand the localization of the items inside the carrier and to interactively change some of the problem constraints directly on the 3D representation.

Keywords: Web3D, logistics, bin-packing

1. INTRODUCTION

The BPP is one of the most relevant optimization problem in logistic and transportation field and represents a classical operative problem for better managing the distribution centres. The main goal is to optimize the carriers utilization, i.e. to minimize the total number of used carriers satisfying some constraints. The common constraints, presented in literature, are on the carriers capacity. Moreover, in some realistic scenarios, other technological constraints are imposed too: load stability, items priority, items overlapping and rotation. From a mathematical point of view, the BPPs are NP-Hard and so the exact algorithms are not used for solving them. On the other hand, the heuristic methods are not able, in some cases, for giving real time solutions because it is very difficult to formalize all the constraints. In fact some relaxations, exact and approximate algorithms have been studied and introduced by Gilmore & Gomory (1965) and by Johnson (1973).

In a general framework, two main entities can be distinguished: the items to be loaded and the transportation carriers, called bins. A bin could be indifferently a vehicle or a container/pallet. The basic unit is not defined by the item but by the wrapping, i.e. the envelope containing the items. It is done for

assuring a more accurate transportation of the items even if it causes some additional costs and weight increments. The interpretation of the BPP results is usually a complex task because it represents the spatial collocation of each item in the bins. Moreover it is often necessary to interact with the solution in order to change some constraint or to retrieve specific data as the item properties. Since our scope of interest is the tri-dimensional BPP, it has been considered to use 3D Graphics as the development tool able to present the results in an intuitive way, allowing the user to query the system and to interactively change any parameters.

3D Graphics is becoming a common tool for the development of advanced user-interface in several application fields. In logistics, for example, there is a wide range of researches that put in evidence the advantages that may be obtained using 3D to support simulation or optimization software. Weznel (2003), proposes a taxonomy of visualization techniques for simulation in production and logistics. He outlines how to use it as a base for decision support to select the right visualization technique for specific target groups.

Bruzzone and Giribone (1996) present one of the first visualization tool that uses together web technologies and 3D graphics as a wired decision support system for logistics.

As regards commercial software for solving the BPPs, two of the most complete ones are Cube-IQ and CubeMaster.

The former gives optimal volume/weight utilization, using its own database, data import/export and reporting. After finding the final configuration, it produces clear 3D load diagrams for seeing it. It also offers full flexibility in loading and stacking rules, which can be defined for each orientation of a box.

CubeMaster optimizes the cargo load on trucks, air & sea containers and pallets quickly and efficiently and supports in planning the order picking, loading and capacity requirements. It allows users to compute the optimal loading diagram and layout, to forecast the best size of orders, to maximize the loading space and weight in cube, and finally to build quickly the reports in 3D graphics.

2. THE BIN PACKING PROBLEM

The mathematical formulations of the BPPs can be defined as linear problems with binary decision variables. In the bi-dimensional problems both items and bins are defined by two parameters individuating a surface. Instead, in the tri-dimensional problems the items have a very low density and so the bins capacity is expressed in term of their volume rather than their weight. For both these problems, we assume a regular shape of the items (also cylindrical or/and composite shapes) and items disposition parallel or orthogonal respect to the bin axis. The items have been classified into three categories for taking into account overlapping constraints: normal, lightweight and heavy items. The bins have been considered with a regular shape and an utilization cost and a capacity are associated to each of them. Our solver considers only bi and tri dimensional bin packing problems under constraints on:

- Overlapping: it is possible to overlap items taking into account their nature (normal, lightweight and heavy) and each item has a maximum weight toleration;
- Rotation: it is possible to rotate the items when specified;
- Priority: when the items have multiple destinations, it could be very important to respect a precise loading sequence (for example items for the same destination have to be loaded near to each other thanks to an interaction among the loading and the itinerary choice (vehicle routing problems)).

Both problems are solved by an approach called Tabu Search Pack (TSPack), i.e. a meta-heuristic that iteratively tries to improve the quality of the current solution (Lodi, Martello and Vigo 2004). This approach, in both cases, has been modified for taking into account new constraints like items rotation and their overlapping. The Tabu Search algorithm starts with an initial solution, obtained considering a total number of bins equal to the number of items and assigning each item to a specific bin, and tries to improve the current solution by applying some feasible moves on the base of the specific problem instance.

2.1. The Tabu Search Algorithm

The Tabu Search Pack (TSPack) is a meta-heuristic applied for improving in an iteratively way the current solution until the optimal configuration is obtained or a maximum number of iterations is performed (Lodi, Martello and Vigo 2004). It is assumed to have n items with a width w_j , height h_j and depth d_j ($j \in J = \{1, \dots, n\}$)), and an unlimited and identical number of bins with width W , height H , and depth D . The priority and overlapping constraints are considered in the internal heuristic A producing a feasible solution for a given problem instance. Let be L a lower bound on the optimal solution and z^* the cost of a feasible solution, if $z^* = L$ the procedure ends; otherwise the search starts generating an initial solution loading each item in a different bin. Some tests have shown that the objective

functions of the BPPs are flat, i.e. many solutions using the same number of bins exist. TSPack introduces a characterization of equivalent solutions based on the fact that a less used bin (target bin) exists and so it is easier to empty it. The target bin is the one that minimizes the filling function :

$$\varphi(S_i) = \alpha (\sum_{j \in S_i} v_j) \setminus V - S_i \setminus n; \quad (1)$$

where $v_j = w_j * h_j * d_j$ and $V = W * H * D$ for 3BP; instead $v_j = w_j * h_j$ and $V = W * H$ for 2BP. S_i is the set of items loaded in the bin i . When the target bin is found, the algorithm tries, through some moves, to empty it changing the loading of a sub-set S of items. S is made up by an item j coming from the target bin plus the ones in the other k bins, where k defines the current neighbourhood. The current configuration is obtained applying the algorithm A to the set S . The important parameters of TSPack are three: K , maximum number of iterations (NitMax) and time limit (TL). The two last ones quantify the algorithm duration (i.e. the number of iterations performed and the total computational time). It means that the TSPack procedure is stopped even if it has not found the optimal solution after a fixed number of iterations or when a time limit is reached. These two parameters are constant during the search and they have to be opportunely initialized (i.e. they should be directly proportional to the number of items n to be loaded). If n is large enough and if the NitMax and/or TL values are not proportionally large, the algorithm could not reach the optimal solution. On the other hand if these values are increased too much, the computational time could be exponential. The most important parameter is K , i.e. the search breadth, and it is independent from the instance and varies dynamically during the algorithm.

The moves applied during the procedure try to empty the target bin changing the loading configuration of the set S . The new solution cost is obtained summing the number of the bins produced by the internal heuristic A applied to the set S and the remaining bins that contains the $\{1, \dots, n\} \setminus S$ items. The solutions obtained are considered feasible if the loading configuration is changed and the total number of bins is not increased respect to the current solution. In other words, the moves try to remove the item j from the target bin without using a new bin.

The internal heuristics A is applied for each item j loaded in the target bin and for each subset of k bins. The value of K is increased by one if the search does not find feasible moves.

If the search finds either a feasible move that decreases the total number of bins or a non-tabu move that loads the set S of items within k bins, it is immediately applied and the value of k (i.e. the dimension of the subset of bins considered in the moves, initialized to 1) is decreased by one. If a move is not immediately applied then a penalty is computed whose value is infinite either if the move is tabu or the internal heuristics uses at least two extra bins. The move

with minimum penalty is chosen if the search does not find any to be immediately applied. When K reaches a maximum value, a diversification procedure is applied.

About the internal heuristic, both for 2BPP and 3BPP, Hybrid Next Fit (HNF) algorithm is considered adapting it for taking into account the capacity constraints for the bins and the priorities, the overlapping and the rotation constraints for the items. After some adjustments, it becomes Hybrid HNF (HHNF). In the 3D case, for example, the routable items are positioned firstly with their shorter side as height in a way that they can occupy the minimum vertical space. Then they are horizontally rotated with the longer side as width. The items are firstly sorted by their specific class (heavy, normal and lightweight) and then by their priority. The HHNF is a section-level-procedure. A section has width equal to the one of the bin and length equal to the longest item inside. A level has height equal to the one of the highest item inside. Then the current item is loaded in the upper left side of the current section if possible; otherwise the section is closed and another is open. If it is not possible, then the item is loaded in the next level. It means that the current level is closed and another is open. If in the bin there is not sufficiently space for loading the item, it is closed and another is open.

3. FRAMEWORK OVERVIEW

The proposed framework it is made up by three main software entities, which can run on separate machines. These entities are the BPP solver, described in the previous section, the Web3D application, and the web interface to the global framework. This latter is a simple JSP page, in which the user can select the order he wants to process, as showed in Figure 1.



Figure 1: The JSP page for the order choice

All the orders information in fact, are stored in a database, and for each order there is the list of the items. The user specifies via web page the order he wants process. Afterwards, the JSP web page retrieve the order information from the database, and send them to the BPP solver. This latter computes the BPP solution and then produces the solution file. Subsequently, the JSP page returns a new html page made up by an applet and a web3d plug-in. Finally, the applet loads the optimized solution previously generated by the solver and stored as file, and dialogues with the visualisation plug-in in order to recreate the 3d scene in accord with

the bin-packing solution. The visualisation plug-in receives the information relative to all the items and the carriers, and then “translate” them in a visual output. This dialogue between the applet and the visualisation plug-in occurs through an Inter-Process Communication (IPC) channel developed for this purpose. All the details about the communication are explained in the next section. In the Figure 2 is represented the temporal diagram of the actions of the different software entities.

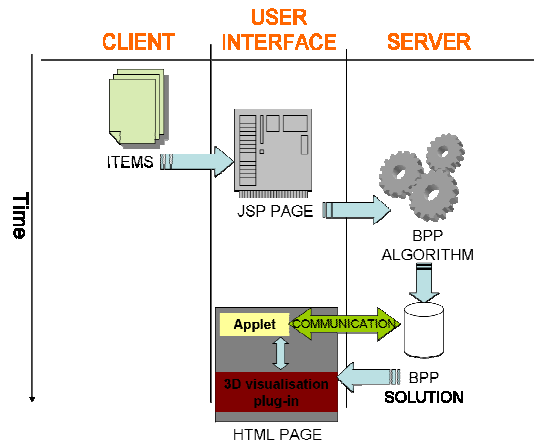


Figure 2: Temporal diagram of the actions of the software entities

The Web3D application has been developed in order to have an easy and flexible interface to interact with the BPP solver, implemented on the web server side. The application offers several functionalities:

1. The optimal BPP solution visualization in 3D graphics.
2. The camera movement to observe the solution from different point of view.
3. The data retrieval about the properties of each item.
4. The movement of the items placed in the optimal solution and the querying of the BPP solver for the feasibility check.
5. The collisions detection during the manual movements of the items.

The final result is the three-dimensional visualisation of the optimal BPP solution for a specific order, as shown in Figure 3. Further, the high level visualisation environment is able to perform advanced visualisation technique not available in the other software. It is possible, e.g., to set the transparency of a 3D entity, like showed in Figure 4.

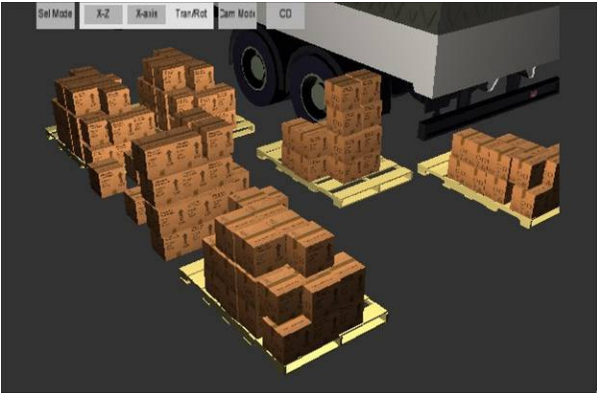


Figure 3: 3D visualization of the solution

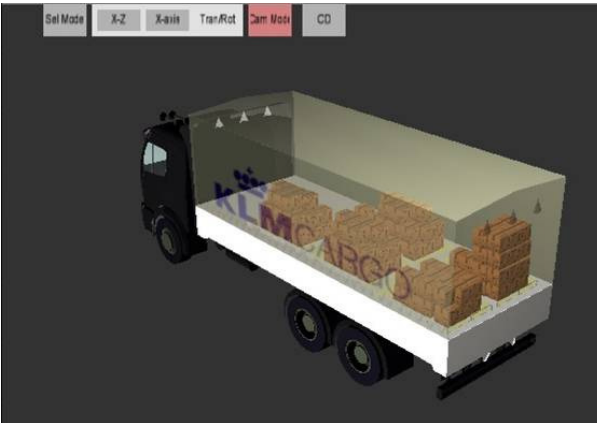


Figure 4: Visualization of items within a transparent carrier

4. SOFTWARE ARCHITECTURE OF THE WEB3D APPLICATION

The Web3D application has been implemented using a commercial software named Virtools Dev (www.virtools.com). It is a complete development and deployment platform for the creation of interactive three-dimensional contents. Using Virtools in fact, it is possible to create a virtual environment by importing 3D objects in several file formats. Further, it is possible to define a behaviour for each object. Virtools, in fact, has its behavioural engine, and it is able to process user events and to execute the behaviour in accord with them. Further, it has its own Software Development Kit (SDK), and therefore it is possible to extend and customize Virtools in order to create new functionalities. As regards our goal, there was the need to interface Virtools with a Bin-Packing solver. In our aims, in fact, the Bin-Packing solver sets the position of the packs within the carriers, and then the visualization of the solution occurs. The communication between BPP solver and the visualisation environment occurs through an inter process communication channel based on network sockets. The final result is a bi-directional communication channel, which allows to set position and orientation of each 3D model parts (both items and bins) in keeping with the optimal solution, and further, to send input data generated by the user to the solver, like the selection or the movement of an item.

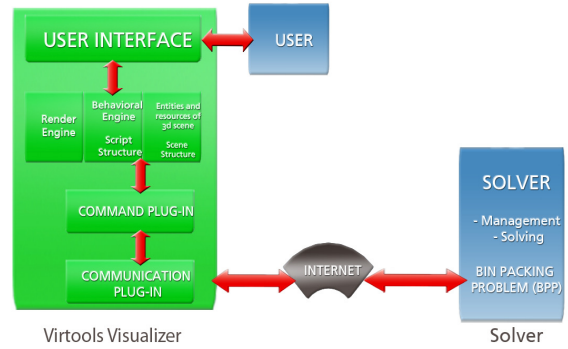


Figure 5: Web3D application architecture

The communication channel is based on a new communication library developed ad hoc, the SimLIB library, which will be described in the next section.

4.1. The SIMLib Library

One of the most common way to achieve the Inter Process Communication (IPC) is using TCP/IP sockets.

However, sockets are a low-level communication channel, and they are not easy to use for the transmission of complex and large blocks of data. Therefore, a new library to simplify the data transmission has been implemented. The new library meets the following needs:

1. Simplification of the use of sockets, since many operations are embodied in macro-functions;
2. Typification of transmitted data: it is possible in fact send data in a structured format;
3. Client-server synchronization: some techniques for synchronization, useful for the application context, have been implemented

First of all, a new data type, embodying all the information to be sent, has been defined, as described in Table 1

Field Name	Type	Description
Operation	Integer	Create, Delete or modify a pack entry
Id	Integer	Id of the object on affected by the operation
Geometry	Integer	Geometry of the entry (0=cube,1=Cylinder, etc. etc.)
Parent	Integer	Id of the parent of the entry
Color	Float	Color in RGB format

	array	
Position	Float array	Position (X,Y,Z)
Orientation	Float array	Orientation in Eulerian angles
Scale	Float array	Scale factor for X,Y,Z dimension.
Description	String	A description of the entry

Table 1: The SimLib_Data data-type

These fields represent the data to be sent in this specific context, i.e. the BPP visualisation. The SimLib_data can be different for another application of the library.

The functions of the library are described in Table 2: SIMLib library functions:

Function Name	Description
SIMServerOpen	Create the server end of a communication channel. This end is waiting for a connection
SIMClientOpen	Create the client end of the communication channel. This end connects to the server.
SimLib_synchroSend	Sends a SimLib_data block through the channel
SimLib_synchroReceive	Receive a SimLib_data block through the client

Table 2: SIMLib library functions

With the help of these functions, it is quite easy sending and receiving SimLib_data block of data through a communication channel. The ends of this communication channel are a SimLib_Client and a SimLib_Server.

4.2. Bi-Directional communication between Virtools Dev and the BPP solver

As previously mentioned, it is possible to assign a behaviour for each 3D entity within the 3D visualisation environment. Behaviours are defined by a set of building blocks. Each building block is responsible of a function. Further it is possible to develop a building block using the Virtools SDK. In particular, three new building blocks have been developed in order to achieve our target:

- **Master Control Block:** It creates a SimLib_Client for the communication and receives data sent by the 3D BPP Solver
- **Action Block:** It receives the SimLib_data block from the MasterControlBlock and, afterwards, it executes a set of operations in accord with data contained in the SimLib_data block.
- **Feedback Block:** It creates a new SimLib_Client to send data to the BPP Solver. As previously mentioned, in fact, the user can modify the solution provided by the solver within the 3D environment in a visual way. The configuration hinted by the user is sent to the BPP solver for the feasibility check through the feedback channel, and afterwards the solver sends the

The final 3D composition created with Virtools, made up by three-dimensional entities endowed with a behaviour, can be exported as Web-Page. In this way it is possible to create complex Web3d applications in a simple and efficient way.

5. CONCLUSIONS

The implemented application is a prototype that shows the advantages of using Web3D technologies as an intuitive user interface for a BPP solver.

The main innovation of our application is the web technologies use for remotely solving a common logistics problem. In this way it is possible to solve logistic problems by using a client machine and querying the solver through the network. Further, the solution is presented in a very effective way using a powerful three-dimensional visualisation environment. Moreover the user can modify the optimal solution directly on the 3D visualisation and check the feasibility and the cost of the new solution.

ACKNOWLEDGMENTS

The research presented in this paper has been realized in the context of the LOGICA project (Regional Technological Laboratory of Logistics) co-funded by Regione Calabria, within the POR 2000-2006 Misura 3.16.

REFERENCES

- Bruzzone A. G., Giribone P., 1998. Decision-support systems and simulation for logistics: Moving forward for a distributed, real-time, interactive simulation environment. *Proceedings of the IEEE 31st Annual Simulation Symposium*, pp. 17-24. April 5-9, Boston, MA, USA.
- Wenzel S., Bernhard J., Jessen U., 2003. A taxonomy of visualization techniques for simulation in production and logistics, *Proceedings of the 35th Winter Simulation Conference: Driving Innovation*. December 7-10, New Orleans, Louisiana, USA.
- Gilmore P.C., Gomory R.E., 1965. Multistage cutting problems of two and more dimensions. *Operations Research* vol.13 pp. 94-119.

Johnson D.S., 1973. *Near-optimal bin packing algorithms*. Thesis (PhD), MIT, Cambridge, MA, USA.

Lodi A., Martello S., Vigo D., 2004. TSpack: A Unified Tabu Search Code for Multi-Dimensional Bin Packing Problems. *Annals of Operations Research*, vol. 131, pp. 203-213.

AUTHORS BIOGRAPHY

Fabio Bruno is an Assistant Professor at the Department of Mechanical Engineering of the University of Calabria. In 2005 he received a PhD Degree in Mechanical Engineering at University of Calabria. He received a Master's degree in Industrial Engineering in 2001. The research activity mainly concerns the application of Virtual and Augmented Reality (VR & AR) to the design process of the industrial products.

Francesco Caruso is a post-doc student at the Department of Mechanical Engineering of the University of Calabria. From 2004 to 2007 he attended his PhD in Mechanical Engineering. His research topics regards Virtual and Augmented Reality, interaction techniques with virtual prototypes and behavioral simulations of virtual prototypes.

Ornella Pisacane is a post-doc student at the Università della Calabria (Cosenza, Italy). Her PhD thesis was concerning with the definition of a simulation-based optimization algorithm for solving the agent scheduling problem in a multi skill call center, supervised by Prof. P. L'Ecuyer (Université de Montréal, Canada) and Prof. R. Musmanno (Università della Calabria). Her current research theme is related to the definition of simulation-based optimization algorithms for efficiently managing supply chains, supervised by Prof. J.F. Cordeau (Université de Montréal, Canada) and Prof. R. Musmanno. She is also working on web services implementation for workflow optimization and for efficiently supporting decision makers related to health care management.

Maurizio Muzzupappa is an Associate professor of Computer Aided Design at the Department of Mechanical Engineering of the Faculty of Engineering of the University of Calabria. From 1989 to 1992, he frequented his Ph.D. at the Department of Mechanical Engineering of the University of Pisa. His current research activities include concurrent engineering (specific topics are the computer support of cooperating virtual engineering teams for design review), collaborative design, virtual and augmented reality and reverse engineering.

Giovanni Venuto was born in Novara (NO), Italy, in 1975. He received his Master Degree in Computer Engineering in 2007. His thesis focused on computer

graphics applied to logistic problems. Actually he works in Pitagora s.p.a. as Software Engineer.

Francesco Rende received his Master Degree in Computer Science at the University of Torino. Then he attended the PhD in Operation Research focused on logistics.

Roberto Musmanno is Full Professor of Operations Research at the Università della Calabria (Unical), Italy. His major research interests are in logistics, network optimization and parallel computing. He has published more than 30 papers in a variety of journals and is coauthor of the book: G. Ghiani, G. Laporte, R. Musmanno, "Introduction to Logistics Systems Planning and Control", Wiley, 2004. He is member of the Scientific Committee of the Italian Center of Excellence for High Performance Computing, funded by the Italian Ministry of University. He is member of the editorial board of the international journal *Computers & Operations Research* (Pergamon) and is chairman of the Council for the Degree in Management Engineering at Unical. He has been involved in several international research projects and was member of the Organizing Committee of international workshops in the field of high performance computing.