Application of Genetic Algorithms to Container Loading Optimization

Raúl Pino, Alberto Gómez, José Parreño, David De La Fuente, and Paolo Priore

***Abstract*—Standardization of transport means, such as, containers has a direct impact on the transportation efficiency sought by European transport policies. In this paper, we present a genetic algorithm application to the container loading problem trying to maximize the cargo volume accommodated in the container whilst ensuring that loading restrictions are met, and thus achieving a reduction in the number of freight to hire and thereby a reduction in costs. The proposed method has been compared to similar models, and the results obtained are similar or even improved.**

***Index Terms*—Genetic algorithms, optimization, container loading**.

I. Introduction

The main goal of the European transportation policies is to promote sustainable mobility through efficient, costly appropriate, safe, environmentally clean and socially accepted transport services. This objective implies an integral concept of the mobility system, enhancing transport nets by using in each section the most appropriate way, optimizing each mode but also the chain as a whole or improving the connections between modes. All this should be supported by advanced information and communication services. This new perspective in the product management has been called "Multimodal transportation".

The multimodal transport of goods is defined as one that uses at least two different transport modes under a single contract of carriage, from a location in a country to another different designated one for delivery and where, typically, a single operator, is responsible for all merchandise management. In this context, an optimal use of the infrastructures and load units along the entire chain is vital. Applying innovative instruments such as those based on new technologies or artificial intelligence make it possible to achieve important benefits to intermodal chain agents:

1) Increasing visibility of the transport chain, given that a document with the exact location of the packages inside the container may be transmitted electronically along with the freight through the different points of the chain.
2) Important savings on shipping costs, as it allows to load a larger amount of goods by optimizing volume occupation in the containers.
3) Guaranteeing a balanced transportation of goods, along with the horizontal and vertical axis. This serve as a protection for the load, for example when it is rocked by waves during maritime transport.
4) Facilitating the goods tasks loading/unloading and its delivery to the clients, taking these factors into account when calculating container optimization.

In this paper, we will try to maximize the cargo volume accommodated in the container whilst ensuring that loading restrictions are met, and thus achieving a reduction in the number of freight to hire and thereby a reduction in costs. In literature this problem is referred to as the "container loading problem" or "bin-packing problem", and it is defined as a problem where "various-sized packages must be packed into several stock containers of certain size .... Since the measurement of the combined packages generally does not equal that of the large containers, one can assume that unused space is left. This too, can be defined as trim loss"[1].

After the seminal study by [2], a number of research works can be found in literature aimed at finding a methodology that leads to an optimal solution for the container loading problem. Some versions include Strip Packing, Knapsack Loading, Bin Packing or Multi-container Loading (see [1], for a general classification, or [3]).

Techniques used for studying the container loading problems range from exact methods or dynamic programming to heuristics including Tabu Search, Simulated Annealing, Greedy Randomized Adaptive Search Procedure (GRASP), Wall-building, Genetic Algorithms [4], [5], etc.

Given the nature of this problem and due to the large number of constraints to consider, we will tackle it through metaheuristic techniques. In our case, we opted for using a Genetic Algorithm.

II. Problem and Proposed Solution

Current research is carried out under the project SITIM, "Analysis, development and evaluation of Intelligent Transport Systems in an intermodal freight environment", supported by Spanish Ministry of Publics Works and whose final objective is the application of intelligent transport systems for freight transport in an intermodal environment, in order to improve effectiveness and sustainability and increase visibility along the whole logistics chain.

One of SITIM participants and the main beneficiary of the proposed system is one 3PL (third party logistics provider) which is a global leader, working with most of the major original equipment manufacturers (OEM). The company is a strong actor in 5 sectors: Automotion, Hi-Tech Electronics, Fast Moving Consumer Goods (FMCG), Healthcare, and

Manuscript received June 22, 2013; revised August 19, 2013. Financial support given by the Government of the Principality of Asturias is gratefully acknowledged.

The authors are with the Polytechnic School of Engineering (University of Oviedo), Campus de Viesques s/n, CP 33204, Gijón (Asturias), Spain (e-mail: pino@uniovi.es, albertogomez@uniovi.es, parreno@uniovi.es, david@uniovi.es, priore@uniovi.es).

DOI: 10.7763/IJTEF.2013.V4.306

Publishing & Media, and run an extensive global network with facilities located in over 100 countries.

One of the main clients in Spain is an important automotive manufacturer. The system has been oriented to this customer, including real characteristics of the spare parts transported in order to reach realistic and practical results. Table I shows the most relevant features of the parts to be handled.

TABLE I: DESCRIPTION OF THE PARTS AND THEIR PACKAGE

PART	*FEATURES*	*PACKAGE DIMENSIONS*	*PACKAGE WEIGHT*	*PACKAGE TYPE*
Engines	*Heavy piece*	*2270×1170×1270*	*1000kg.*	*Returnable metallic package*
Gearboxes	*Heavy piece*	*2270×1170×1090*	*700 kg.*	*Returnable metallic package*
Different parts	*Different features*	*Standard package*	*Variable*	*Lost package*
Special parts	*Fragile or special geometry*	*Special dimensions*	*Variable*	*Non-standard package and non-returnable*
Parts in returnable packages	*Different features*	*2270×1170×1270 or 2270×1170×1090*	*Variable*	*Returnable metallic package*
Small-size parts, regrouped	*Different features*	*Standard package or 2270×1170×1270 or 2270×1170×1090*	*Variable*	*Returnable metallic package or cardboard*
Small box to introduce in other packages	*Different features*	*Small size*	*Variable*	*Inside a metallic package already counted*

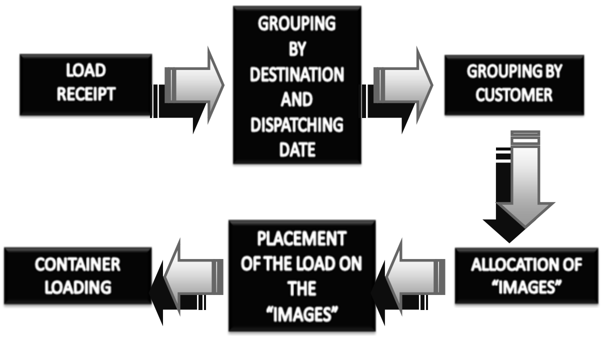

Fig. 1. Current process of container loading.

Current container loading process is depicted in Fig. 1. The process begins with the receipt of the material to be transported. This material has been previously packaged using the appropriate packaging: metal returnable, non-returnable, etc. As the goods are received, the corresponding dispatch labels are produced so that the packages are well identified (content, delivery date, dispatch date, customer and destination).

Then a first grouping is made according to the destination and dispatching date. Afterwards a second grouping is carried out by customer. If the volume to be transported for each customer is enough to fill one or more containers, the allocation of containers will be made by customer. Otherwise, there will be multi-client containers provided their destination is the same.

Aiming at reducing the time needed for completing the container loading process and optimizing the packaging placement within the container, a so-called "image" of the container is developed: that means that an employee draws an area perimeter on the facility floor with the exact dimensions (length and width) of the container bottom. A bar will be used to set the height of the container.

Then, the operator starts filling the "image" with the different packagings to be loaded in the container, similarly as if he would be filling the real container. This task is performed on the basis of his experience and trying to meet the existing restrictions regarding the maximum allowed height, maximum allowed weight, distribution of weights, stacks of packaging, etc.

Going in deeper detail, the restrictions to be met can be summarized as follows:

1) Maximum container weight and load balance. The maximum weight allowed in a 40 ft. container is 26,460 Kg. and it should be evenly distributed in the container.
2) Packaging Stacking:

- According to the height of the container, packaging may be stacked in 2, 3 or 4 levels.
- Packaging weight. There is an packaging encoding by weight: "B" code for the heavy packages (weight >500 Kg; B packages must be placed at the bottom of the container); "M" for medium (weight between 250 and 500 Kg.; M packs will be placed whether at the bottom or on top of B packages); and "H" for low weight (<250 Kg; H packs shall be located at the top of the container).

TABLE II: EXAMPLES OF STACKING

Top Position	H	M	H	B	H	M	H	B	M	B
Middle Position	M	H	B	H	H	M	B	H	B	M
Low Position	B	B	B	B	M	M	M	M	H	H
Allowed?	Yes, desired result	No	Analyze middle load	No	Yes	Yes, analyze middle and top load	No	No	No	No

Table II shows some examples of allowed and not allowed stacking.

Other criteria are:

- Accumulated weight of the stack. Each package will support a maximum weight of 750 kg. The weight of H packs cannot exceed 250 kg.

- Fragility of the parts. Fragile packs will be placed at the top.
- If the packaging was not completely filled in, it will be placed in upper positions.

The formation of the images is done from the back towards the front (Fig. 2), since the containers are loaded by entering the goods through the back door. Therefore the first packages to be placed in the container will be those placed at the front of the image and consequently, those last placed in the image.

Our main contribution is the development of a computer program based on genetic algorithms, which helps in the process of the images preparation. As previously mentioned, this process is currently carried out manually, relying on the knowledge and mainly the experience of the operators responsible for the process. The objective is to reach a minimum 65 m^3 occupation in a High Cube (40 ft.) container which implies to fill the 85% of the total volume of the container.

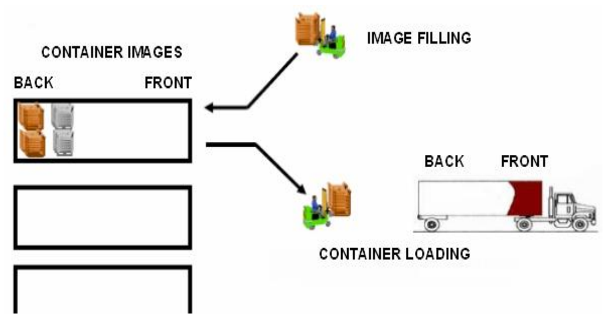

Fig. 2. Formation of images and the subsequent container loading.

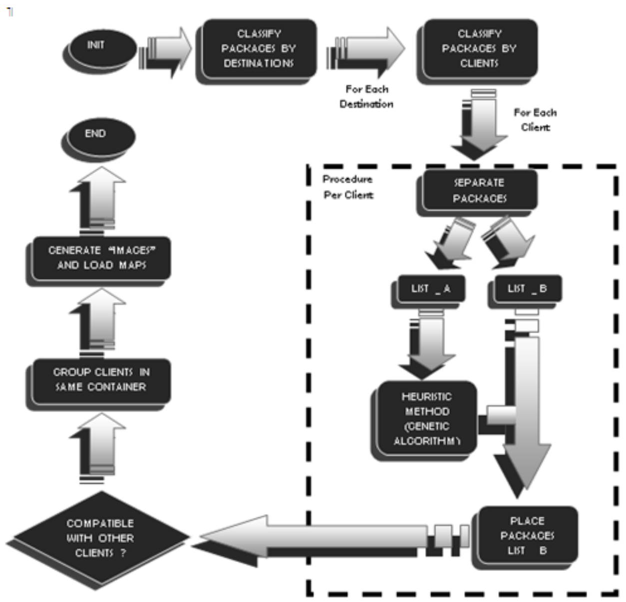

Fig. 3. Proposed procedure.

The methodology proposed in this paper provides the package filling sequence for the container image so that the container volume utilization is maximum. Additionally, the solution can be spatially visualized in three dimensions. The general procedure followed for calculating the filling of the container is described in Fig. 3. The objective is to load all the freight into the minimum number of containers possible, taking into account the mentioned restrictions.

The first step in the process is, once the freight to be disposed for a certain date is available; separate it by destinations, given that a container must be filled with packages for the same destination. After this, given a fixed destination, the procedure is applied for each client, in order to keep their freight grouped. Desirably, each container should store the freight of only one client, but usually the load is not large enough for filling a full container, so in the last step, the process tries to create "compatible containers". The procedure per each client can be summarized as follows: Packages are divided in two lists. List A contains the more bulky and heavy ones and List B the smaller and lighter, according to percentiles. Packages from List A are applied a heuristic method (genetic algorithm which is described in next sections) for optimizing container occupation in 3D. The result is a list with the sequence of packages as they should be introduced in the container, and its objective function takes into account the mentioned restrictions on load balancing and spatial occupation of the container, with higher or lower weight on the whole value of fitness. These restrictions are incorporated into the objective function through the area of the surface already covered, the total volume occupied by the packages and the imbalance detected both in *X* and *Z* axis. The function that determines the goodness of the solution for each container is:

$$F=\begin{cases}\frac{1}{c}\cdot\left[\sum_{i=1}^{c} w_i\cdot Volume(C_i)+w_i\cdot Area(C_i)+\right.\\ \left.+w_i\cdot\left[1-\left(\frac{Diff(Z\_C_i)-Diff(X\_C_i)}{2}\right)\right]\right] \\ ;ifDiff(X\_C_i)\le w\_weight \\ 0;ifDiff(X\_C_i)>w\_weight\end{cases} \quad (1)$$

being:

- *C* the total number of containers used ($C \in$ [1, *total_number_of_container*]).
- *w_weight* the maximum difference in weights tolerated for guaranteeing the balancing of C_i (*w_weight* $\in$ [0.1]).
- w_i, corresponds to the weights assign to the total area occupied, to the total volume and to the imbalance of the container, according to the importance given to each factor on the global goodness of the solution ($w_i \in [0, 1]$):

$$\sum_{i=1}^{n} w_i = 1 \quad i \in I = \{1,..,n\} \quad (2)$$

- $Area_{Ci}$ the area occupied by the packages in relation to the total area of the container *Ci* ($Area_{Ci} \in [0, 1]$).
- $Volume_{Ci}$ the volume occupied by the packages contained in C_i in relation to the total volume of the container ($Volume_{Ci} \in [0, 1]$).
- $Diff\_z_{Ci}$ the difference in weights between the two halves of the container C_i along the Z axis ($Diff\_z_{Ci} \in [0, 1]$).
- $Diff\_x_{Ci}$ the difference in weights between the two halves of the container C_i along the X axis ($Diff\_x_{Ci} \in [0, 1]$).

A fitness penalty exists that increases in proportion to the difference in weights between the two halves of the container; $Diff\_zC_i$ = $Diff\_xC_i$ = 0 corresponds with perfect balance. Moreover, the maximum penalty is applied (goodness equal to zero) if the imbalance exceeds the threshold *w* weight along the width of the container (*X* axis). This threshold is more permissive for imbalance along the length of the container (*Z* axis), but more strict when it comes to its width. After applying this method to list A, packages from list B will be used to fill the existing gaps. This second approach of filling the gaps with small packages is a combination between the methods proposed by [6], and [7].

Finally, if a container results with a low occupation, the process tries to create a multi-client container, provided the destination is the same and the addition of both clients in the same container does not exceed its total weight and volume limits.

III. Placement Strategies

This section lists the different strategies followed for optimizing the space when placing the packages of list A through the heuristic method briefly explained in previous section. These strategies experienced an important evolution during the development of the algorithm, from the initial and simplest strategy to the one adopted at the end:

- *Strategy 1.* The first strategy is responsible for the placement of the packages of list A at the base of the container (Level 1), assigning its final position through the movement of the box on the surface of the container with successive displacements to the bottom and to the left side of the container, following the 2D method implemented in [8]. When one of the packages cannot be allocated in the free space of the container or its weight added to the total weight of the container exceeds the maximum allowed payload, a new container is created to allocate this box and the following.Obviously, as this method only applies the heuristic technique to the base of the container, equivalent to a 2D-optimization technique, it is highly difficult to achieve an optimal solution in terms of volume and load balancing, being necessary to extend the strategy to a higher number of levels.
- *Strategy 2.* As an improvement to the previous strategy, an attempt to place the current box over the previous one placed was introduced, thereby establishing a second level of packages. Thus, a box can be placed over another as long as it does not exceed the maximum weight limit supported by the pile of packages and provided its dimension is smaller or equal to the dimension of the base box. In case a box cannot be placed in Level 2, the strategy tries to located it on Level 1, through the described displacements on the surface of the container. Regarding the creation of a new container and the placement of packages from list B, this strategy operates in identical way to the previous one.
- *Strategy 3.* It is identical to the previous strategy, except that the attempt to place the current package at Level 2 is carried out on all the packages already placed at Level 1, not only on the previous one, increasing placement opportunities but also the complexity of the algorithm.
- *Strategy 4.* This strategy extends the previous one to *N* levels. Firstly, it tries to place each box over any of the packages located at Level 1. If this cannot be achieved, it searches among the packages at Level 2, Level 3, and so on. The set of packages on which it attempts to place the current box is restricted to the immediately preceding in the sequence that forms a stacking. Finally, if it is not able to place the package on anyone else, it tries to place it on the surface of the container.
- *Strategy 5.* It works like the previous strategy, but the set of candidate packages to place the current one is not restricted only to the immediately preceding, but to all the packages of the container. Thus, it iterates through all the packages from the first placed until the previous to the current one.
- *Strategy 6.* Up to now, the exposed placement strategies are restricted to a single container. This strategy is identical to the previous one, except that it tries to search for a candidate package in all the containers, following the sequence from the current one to the initial container (C_n, C_{n-1}, C_{n-2}, ..., C_1). If it is unable to place the box in any of them, a new container is created. Moreover, the technique of moving the packages on the surface of the container is complemented by the generation of a set of points for placing them (similar to the method described in Section 5.2), which helps to put the box directly to its final location or, in the worst case, into a place that needs very few corrections. Furthermore, the placement of the box on previous packages is optimized, avoiding the need of displacing it in case the base box has no another box over it.

In order to calculate the goodness of the solution (the value of its objective function) for the strategies described, equation (1) is applied. Specifically, the weights assigned are w_1 = 0.75, w_2 = 0 and w_3 = 0.25.

The results obtained relative to the goodness (1) and maximum occupation of containers is shown in Table III. The total number of packages used as input, is 160, divided in 6 different types (sizes and weights).

As it is shown in Table III, the best value is obtained by Strategy 6 (85.5% of volume occupation), as it tries to find the perfect place for the box among all the existing containers.

TABLE III: Results Obtained for Each Strategy

Generation Type	*Goodness*	*Max. Occup. (%)*	*Balance X*	*Balance Y*
Strategy 1	*0,459*	*37,26*	*3,38*	*0,91*
Strategy 2	*0,564*	*46,37*	*0,11*	*1,66*
Strategy 3	*0,670*	*54,63*	*0,86*	*0,20*
Strategy 4	*0,674*	*52,01*	*1,10*	*2,26*
Strategy 5	*0,674*	*50,81*	*0,41*	*0,80*
Strategy 6	*0,891*	*85,50*	*0,27*	*0,04*

Finally, the proposed method for the placement of the packages that belong to List B, is inspired by the one presented by [6], which generates a set of possible points where a box can be located, taking as reference the already located packages (both from List A, whose location is calculated previously by the heuristic technique, and from List B, while they are introduced). These points represent possible locations for the remaining packages, provided they not exceed the maximum payload of the container nor the maximum weight that can be supported by the base box (if the current package tries to be placed at a level higher than Level 1.

It should be mentioned that, after each placement of a new box and the generation of a new set of points, it is necessary to carry out a new analysis of the entire set of points, responsible for adjusting the dimensions and weights that the points are able to sustain in case of overlapping with other points or with the packages already placed. Furthermore, when selecting candidate points where to place the box, taking into account the restrictions mentioned above, a preliminary calculation of how the placement of the box affects load balancing is made, restricting the selection of points to the half of the container where the limit of imbalance tolerated (5%) is not exceeded. If none of the halves of the container exceeds the threshold, all the existing points are taken into account. Given a set of points where a box from list B can potentially be placed, is taken as the final location that point which has a lower value of *X*, lower value of *Y* and lower value of *Z* (in that order), between those who belong to the half of the container in which the introduction of the new box does not unbalance the load above the threshold.

IV. Results

In order to demonstrate that the procedure presents adequate results in volume terms, it has been compared to the methods proposed by [9] and [10]. The results are in Table IV.

TABLE IV: Comparative with Proposed Methodology

Test Case	*Volume Utilization [%]*				
	Ratcliff and Bischoff	*Davies*	*Proposed Solution*		
			Mean	*Min*	*Max*
BR1(3)	*75.56*	*78.99*	***82.95***	*71.50*	*92.45*
BR2(5)	*78.74*	*83.04*	***85.02***	*73.47*	*91.83*
BR3(8)	*80.91*	*84.62*	***86.44***	*78.42*	*93.52*
BR4(10)	*80.96*	*84.69*	***86.44***	*77.99*	*90.93*
BR5(12)	*80.35*	*83.73*	***86.38***	*82.02*	*92.45*
BR6(15)	*79.90*	*84.12*	***85.89***	*80.89*	*89.95*
BR7(20)	*78.93*	*82.74*	***85.56***	*79.53*	*91.80*

The number in parentheses in the first column in Table IV represents the number of different types of packages in each test, for a total of 100 packages. It can be appreciated that with increasingly heterogeneous orders, our algorithm works slightly better than theirs. BR1 shows the worst results, but it only operates with three types of boxes, which is not common in real orders.

V. Conclusions

This paper has described the implementation of a genetic algorithm for solving bin-packing problem, applied to a real case study. The main objective is to distribute the freight of several clients into the minimum number of containers meeting several restrictions related to stacking and balance aspects. The procedure followed combines a heuristic method (genetic algorithm) for the heaviest packages with another based on [6], that fills space gaps with the smallest boxes, achieving important reductions in execution time. Through the analysis of the most suitable GA operators, being necessary to execute a high number of tests and selecting the average values, a best configuration has been presented. This configuration of the algorithm meets all the requirements in a reduced time (less than 20 seconds) and even improves the initial objective (85% of total occupation) for an input order which simulates a conventional order in a real context.

References

[1] H. Dyckhoff, "A typology of cutting and packing problems," *European Journal of Operational Research*, vol. 44, no. 2, pp. 145-159, 1990.

[2] P. C. Gilmore and R. E. Gomory, "Multistage Cutting Stock Problems of Two and More Dimensions," *Operations Research*, vol. 13, no. 1, pp. 94-120, 1965.

[3] G. Wäscher, H. Haussner, and H. Schumann, "An improved typology of cutting and packing problems," *European Journal of Operational Research*, vol. 183, pp 1109-1130, 2007.

[4] A. Bortfeldt and H. Gehring, "A hybrid genetic algorithm for the container loading problem," *European Journal of Operational Research*, vol. 131, no. 1, pp. 143-161, 2001.

[5] H. Gehring and A. Bortfeldt, "A parallel genetic algorithm for solving the container loading problem," *International Trans. in Operational Research*, vol. 9, no. 4, pp. 497-511, 2002.

[6] S. Martello, D. Pisinger, and D. Vigo, "The three-dimensional bin packing problem," *Operations Research*, vol. 48, no. 2, pp. 256-267, 2000.

[7] W. Huang and K. He, "A caving degree approach for the single container loading problem," *European Journal of Operational Research*, vol. 196, no. 1, pp. 93-101, 2009.

[8] A. Gómez and D. D. L. Fuente, "Resolution of strip-packing problems with genetic algorithms," *Journal of Operational Research*, vol. 51, no. 11, pp. 1289-1295, 2000.

[9] M. S. W. Ratcliff and E. E. Bischoff, "Allowing for weight considerations in container loading," *Operations Research Spektrum*, vol. 20, pp. 65–71, 1998.

[10] A. P. Davies, "Approaches to the container loading problem," Unpublished Ph.D. dissertation, University of Wales Swansea, 2000.

Raúl Pino graduated in Industrial Engineering in 1992 at University of Oviedo, Spain and gained his Ph.D. from the University of Oviedo in 2000. He is currently a professor at the Gijón Polytechnic School of Engineering. His current research interests are in the areas of artificial intelligence, forecasting, simulation and manufacturing. He has published a great number of research papers in a number of leading journals.

Alberto Gómez works for the Department of Business Administration and Accountability, the School of Industrial Engineering of The University of Oviedo, Spain. His teaching and research initiatives focus on the areas of production management, applied artificial intelligence and information systems. He has written several national and international papers. Journal of the Operational Research Society. Artificial Intelligence for Engineering Design, Analysis and Manufacturing. International Journal of Foundations of Computer Science. European Journal of Operational Research. International Journal of production economics. Engineering Applications of Artificial Intelligence. Concurrent Engineering-Research and Applications.

Jose Parreño graduated in industrial engineering in 1991 at University of Oviedo and gained his Ph.D. from the University of Oviedo in 2002. He is currently a professor at the Gijón Polytechnic School of Engineering. His research interests include forecasting and artificial intelligence applications in production problems, simulation, and manufacturing. He has published research papers in a number of leading journals such as: Journal of Mechanical Engineering, Engineering Applications of Artificial Intelligence and Applied Artificial Intelligence.

David De La Fuente is a full professor on Operations Management & Management Science. He got the Ph.D. on Management Science in 1987 in Universidad de Oviedo. He was the vice dean in Industrial Engineering School, University of Oviedo, Spain, from1987 to 1991. He is a invited professor at several Latin-American Universities in Bolivia, Argentina, Cuba, El Salvador etc. He is also a member of the International Scientific Committee of 30 Technical Conferences. David is the director of several research projects (European, national and regional), and contracts with companies. He has published 20 papers in international journals indexed by JCR and SJR. He is a member of POMS, EurOMA and INFORMS, etc.

Paolo Priore graduated in industrial engineering in 1991 at University of Oviedo and gained his Ph.D. from the University of Oviedo in 2001. He is currently a professor at the Gijón Polytechnic School of Engineering. His research interests include machine learning applications in production problems, simulation, and manufacturing. He has published a great number of research papers in a number of leading journals such as: Applied Soft Computing, Engineering Applications of Artificial Intelligence, Applied Artificial Intelligence, Computers & Industrial Engineering, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, etc.