SPACE OPTIMIZATION IN WAREHOUSES LOGISTICS

Francesco Cepolina[(a)], Elvezia Maria Cepolina[(b)]

[(a)]DIME University of Genoa
[(b)]DICI University of PISA

[(a)]fcepolina@hotmail.com, [(b)]e.cepolina@ing.unipi.it

ABSTRACT
Industrial warehouses are typically composed by multi levels scaffolds holding pallets, each pallet carrying one or more manufacturing item. The paper introduces the problem of the wasted space in industrial warehouses and describes the algorithms implemented in a simulation software able to suggest how to organize the storage of the manufacturing items. The proposed software uses effective and efficient search algorithms to minimize the overall storage lost space. The general structure of the numeric platform and the logics of the solver algorithms are described together with the strategies adopted to reduce the computation time. Typical results obtained in an industrial environment working in the plastics sector are presented and discussed.

Keywords: warehouse management, space optimization, palletized objects, logistics, multi stages optimization

1. INTRODUCTION

One of the main tasks of manufacturing plant organizations is to reduce the storage space dedicated to tools, fixtures and devices and to improve their inbound logistics efficiency by minimizing the displacements and optimizing the positions. The size and weight of the objects to be handled in the production plants can vary widely. Generally sets of these objects are carried by pallets that are displaced using electric lifts; the forks of the electric lifts can quickly load and unload the pallets. The preferred solution is to use ISO standard pallets of the same size in order to facilitate the automation. In effect today many new storages are fully automated (ASRS) and can host several thousands of pallets.

The objects within a warehouse are usually subdivided in categories; each section of storage hosts a different typology of products and tools. It is not always trivial to organize the objects in order to minimize the warehouse wasted area; the correct space optimization allows reducing the overall number of pallets needed to carry goods, allowing maximizing the overall number of objects that a store can hold.

The storage space optimization needs come directly from the industry (Chyuan et al. 2009), the analytical methods used to solve this problem belong to the field of applied research. A review of the optimization methods in shelf space allocation is given in (Bai 2005).

The physical constraints and integrality constraints of the warehouse space allocation problem are very similar to the constraints in bin packing and knapsack problems, which are well-known NP-Hard problems (Martello and Toth 1990a). However, warehouse space allocation problem may be even more difficult because it usually has a non-linear objective function and some additional constraints e.g. weight constraint has to be considered.

In the literature, optimization studies concerning warehouse space allocation problem have been carried out: appropriate models have been developed and optimization techniques proposed. A frame of optimization approaches that were considered by the authors and gave rise to the original approach presented in the paper is hereafter given.

Optimization approaches include exact and heuristics methods.

An exact method seeks the optimal solution to the problem. Exact methods include well-known linear programming (Hillier and Lieberman 2005), dynamic programming (Bellman 1957), branch and bound (Hillier and Lieberman 2005), and Lagrangian relaxation method (Reeves 1995). Although these approaches could obtain optimal solutions, they can be computationally expensive and impractical for many real-world applications.

Heuristics could be used to create a solution or to improve an existing solution by exploring the neighboring solutions based on given appropriate rules or strategies (Reeves, 1995), but these simple heuristic methods are prone to getting stuck in a local optimum. To prevent this risk, many researchers proposed advanced heuristic approaches, called meta-heuristics (Osman and Kelly 1996; Glover and Kochenberger 2003) including single point, population based and hybrid methods.

In the following, references will be given about space allocation problems similar to the one considered in the paper.

The one dimensional bin packing problem refers to a given set of items $I = \{1,..., m\}$ each having an associated size or weight w_i and a set of bins with identical capacities c. The problem is to pack all the items into as few bins as possible, without exceeding the capacity of the bins. This packing problem belongs to the family of NP-Hard combinatorial optimization problems (Martello and Toth 1990b) and there is no known polynomial time-bounded algorithm that can solve every problem instance to optimality. This problem can also be extended to two-dimensional and three-dimensional bin packing, where both the bin and the items have sizes in two or three dimensions. The one-dimensional bin packing problem has been addressed by many researchers and both exact methods and meta-heuristic methods have been applied.

The knapsack problem is similar but the variables weight and profit are introduced for each item. The problem is to select a subset of items such that the total profit of the selected set of items is maximized. Typically exact approaches such as branch and bound algorithm (Martello and Toth 1975) and dynamic programming (Toth 1980) are used. However, due to their significant computational requirements, for very large domains (m>1000) approximate approaches are preferred.

The problem of storage space optimization is non linear and can be solved adopting a heuristic approach following the principles of the dynamic programming (Bellman 1957). The problem is decomposed in nested stages, each corresponding to a sub-problem in only one variable that must be solved before moving to the second stage. The problem can hence be tackled by a recursive procedure, in which each iteration corresponds to a sub-problem. Moving from the first stage to the final one, the model will provide an optimal answer.

As said before, optimization problems are highly computationally expensive when dealing with large size item domains (Hillier and Lieberman 2005). In such circumstances one may refer to some approximation methods which can solve the problems with satisfactory solution quality within reasonable computational time, resorting to heuristic and meta-heuristic approaches. In our case a memory meta-heuristic method using a short term history, including iterative local search, is adopted.

Wasted area of the shelf that is expensive due to the high costs of construction as well as maintenance.

In the paper the problem of storage space optimization has been approached in sequence in two stages: vertical optimization and horizontal optimization.

The first problem deals with the height of items and aims to assess the vertical distance between the shelves: adjustable shelving systems allow varying the distance between the shelves. The vertical wasted area is minimized putting, under the same shelf, moulds having all almost the same height.

The second problem deals with width and length of items and aims to minimize the number of required pallets, and therefore the wasted area on each pallet.

The paper has been structured in the following way: section 2 describes the problem constraints. Section 3 describes the vertical optimization and the algorithm proposed to solve it. Section 4 focuses on the horizontal optimization and on the algorithms that have been developed to solve it. The algorithms have been applied to an illustrative case of study that is presented in section 5. Conclusion follows.

2. BASIC CONSIDERATIONS

In the following the optimization of a specific industrial moulds storage is considered as a reference case; the problem is typical of important industrial sectors such as plastics, shoes, clothing, automotive. The studied warehouses can host over 500 moulds and each year new moulds are added to the warehouse. A good storage exploitation allows saving space and enhances the moulds picking speed as well as the manufacturing plant internal logistics.

The pallets considered belong to the same ISO family and have the same area layout; this hypotheses is widely accepted because it facilitates the use of automated warehouses. Each pallet can accommodate a number of moulds that ranges from 1 to 6, according to their dimension.

The shape of each mould is approximated, by hypotheses, to the enclosing minimum parallelepiped: specific algorithms can be implemented if the objects to place on the same pallet have complex geometries (Schepers and Wottawa, 1996).

Each parallelepiped mould, by hypothesis, can be positioned on the pallet only with bottom edges parallel to the pallet sides; it can be rotated of 90 degrees around the Z axis assuming two possible footprint configurations on the pallet outline (see figure 1).

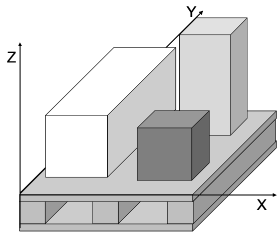

Figure 1: Pallet holding some items

The moulds that can be accommodated on a given pallet has to satisfy constraints on the overall weight of the pallet and on the overall area of the moulds (that should not exceed the pallet area). Only one layer of moulds is admitted on a given pallet.

Moreover nested configurations of moulds on a given pallet, as shown in figure 2, are not allowed.

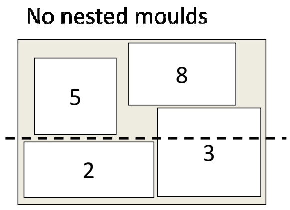

Figure 2: Nested moulds

The displacement of heavy moulds is a delicate operation: nested moulds are forbidden to simplify the factory activities. The condition “no nested moulds” allows the workers to cross the moulds horizontally. The moulds, that weight from 20 kg to 300 kg, need to be handled with care.

The shelving system allows varying the distance between the shelves. A suitable low air gap is needed for pallet loading unloading operations (see figure 3). The depth of the shelves allows to accommodate only one pallet. All the moulds should be easily accessed and no priority in mould loading/unloading operations has been taken into account.

3. VERTICAL OPTIMIZATION

Moulds are clustered according to their heights: the moulds are divided in groups. A given shelf accommodates only the moulds belonging to a group. The distance between shelf *n* and shelf *n+1* depends on the max height of the moulds within the group assigned to shelf *n*, as shown in figure 3.

We call *m* in the following the number of moulds belonging to a given group that should be assigned to pallets.

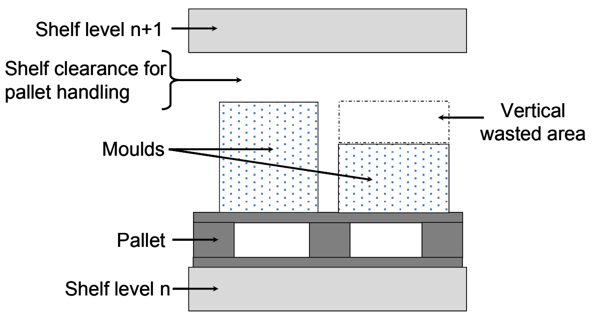

Figure 3: Vertical wasted area

4. HORIZONTAL OPTIMIZATION

The target is to assign all the moulds belonging to a given group to a minimum number of pallets.

The algorithm proceeds by successive steps. A threshold *TH* (that represents the maximum wasted space that we accept on one pallet), is fixed in the process initialization (*TH°* could be for example 10%) and it is sequentially increased (in the external iterations). For a given value of *TH*, the algorithm tries to assign a number *n* of moulds to the pallets. *n* increases from 1 to 6 as the internal iterations increase. Given *TH* and *n*, the algorithm fills as much pallets as possible with *n* moulds sets: for each pallet the algorithm (n MOULDS/PAL procedure deeply described in the section 4.1) finds the *n* moulds, out of the *m* available in the given group, and their orientations (virtual pallet) that satisfy constraints at section 2 and, in the cases of the first two approaches proposed in sections 4.1.1 and 4.1.2, minimize the wasted area on the pallet. If the virtual pallet, output of n MOULDS/PAL procedure, satisfies also the constraint on *TH*: *the wasted area on the pallet<=TH,* the *n* moulds, oriented as they are in the virtual pallet, are assigned to the current pallet.

The proposed methodology is schematically described in the following. The architecture of the solver is shown in figure 4.

1. TH=critical threshold:=TH°
2. p:=1
3. n:=1
4. n MOULDS/PAL problem →finds the n moulds (out of the m available in the given group)and their orientations (virtual pallet) :
   a. that satisfy constraints at section2
   b. (that minimize the wasted area on the pallet)
5. check if the wasted area in the current virtual pallet is lower than TH
6. if yes
   a. assign the n moulds to the pallet
   b. delete the n moulds from the group →m:=m-n
   c. p:=p+1: go to the next pallet
   d. go to step 4
7. if not
   a. n:=n+1increases the number of moulds to be assigned to the pallet
   b. go to step 4
8. is m=0? Have all the moulds of the given group been assigned to pallets?
9. if not
   a. TH:=TH+10%TH
   b. Go to step 2
10. if yes→END

A multistage solver has been developed and implemented. The core of the algorithm is dedicated to the search of the *n* moulds and their orientations that satisfy constraints in section 2 and, as it concerns the first two approaches proposed in sections 4.1.1 and 4.1.2, minimize the wasted area on the pallet.

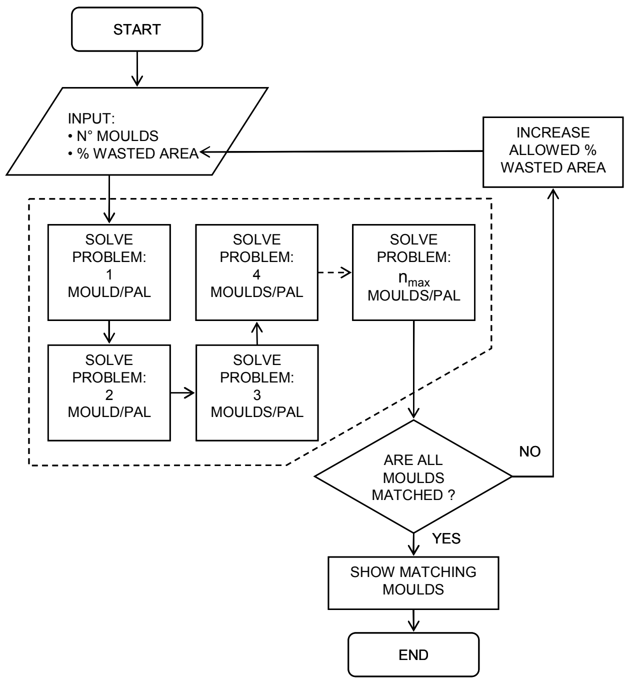

Figure 4: The architecture of the solver

Proceedings of the Int. Conf. on Harbor Maritime and Multimodal Logistics M&S, 2014
ISBN 978-88-97999-39-3; Bruzzone, Del Rio Vilas, Longo, Merkuryev, Piera Eds

4.1. The problem: n MOULDS/PAL

The number n of moulds to be assigned to each pallet is given in input to this procedure. The procedure aims at finding the n moulds, out of the m available in the given group, and their orientations (virtual pallet) that satisfy constraints at section 2 and, in the cases of the first two approaches proposed in sections 4.1.1 and 4.1.2, minimize the wasted area on the pallet.

The dimension of the search space is: $= \frac{m!}{(n)!(m-n)!} 2^n$ where:

- $(\frac{m!}{(n)!(m-n)!})$ is the number of combination of m moulds on n positions without repetition and,
- 2^n takes into account that each mould can be loaded on the pallet with 2 orientations.

As an explanatory example we take into account the case of $m=10$ moulds and $n=4$ moulds on a pallet Each mould, out of 10, can be positioned, on a pallet area; the 4 moulds can be horizontally (X) or longitudinally (Y) oriented respect to the pallet outline. It is therefore possible to load the pallet with $\frac{10!}{4!6!} 2^4 =$ 3360 possible configurations.

Three alternative approaches have been proposed and they are schematically represented in figure 5. The first two aim at finding the configuration (virtual pallet) that satisfies the constraints and minimizes the pallet wasted area. The third approach aims at finding a virtual pallet that only satisfies the constraints. All the three approaches have been tested and are reported in the following.

All the three approaches are based on a solver that hosts internally two generators.

The first solver generator (RELATIVE POSITIONS in figure 6) creates all the possible moulds orientations (X,Y) on the pallet. *2m* objects are generated from the *m* moulds.

Then the second solver generator (SET CONFIGURATIONS GENERATION) creates possible combinations of the *2m* objects (virtual pallets).

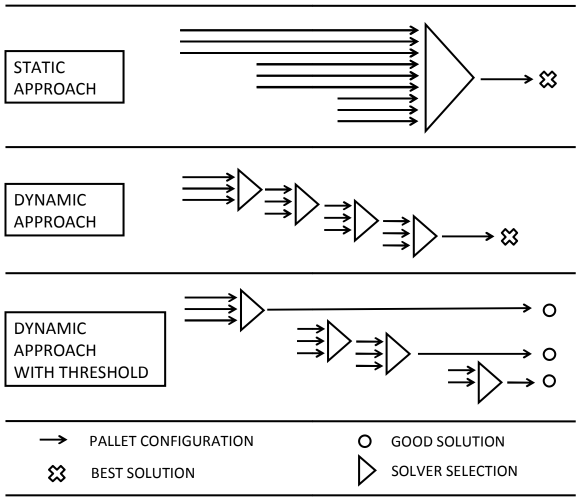

Figure 5: Open loop static and dynamic approaches

4.1.1. Static approach

The second solver generates all the possible virtual pallets (which number is $\frac{m!}{(n)!(m-n)!} 2^n$). Then a check on all the constraints described in section 2 is performed (MATCH CRITERIA in figure 6) for all the generated virtual pallets: all the virtual pallets that do not satisfy the constraints are deleted.

A performance parameter is then assigned to all the remaining virtual pallets. The performance parameter is related to the wasted area: the area of the pallet that is not covered by moulds. The solver finds the virtual pallet that has the best performance parameter (SEARCH OF THE BEST SOLUTION in figure 6).

At the end of the procedure the set of moulds belonging to the best virtual pallet is sent to the check (step 6 of the proposed methodology) on *TH*.

The five basic steps, shown in figure 6, summarize the solver activities.

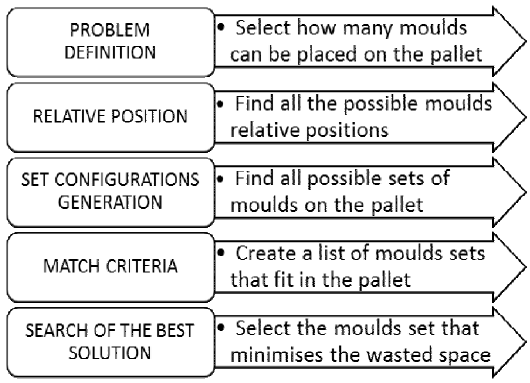

Figure 6: Logical steps of the solver activities

This approach tests all the possible combinations by numeric simulation. This approach is appealing but an important aspect needs to be considered; the number of solutions to explore can be really high.

4.1.2. Dynamic approach

The solver, addressed in the previous paragraph, requires a huge computational effort: it needs about one

week to solve the problem $m=100$ moulds and $n=6$ positions. Some simplification hypotheses are now introduced in order to speed up the computation time: the logical steps to solve the problem are critically reviewed and improved.
In the dynamic approach, the second solver proceeds by sequential steps. At each step *nb* (not all, as in the static approach) virtual pallets are generated, a check on the constraints is performed (MATCH CRITERIA in figure 6) on the *nb* generated virtual pallets: all the combinations that do not satisfy the constraints described in section 2 are deleted. The performance index is assessed for each of the remaining virtual pallets and the best one is selected. The best solution is added to the *nb* combinations generated in the next step. This procedure is repeated until all the possible virtual pallets are taken into account.

At the end of the procedure the set of moulds belonging to the best virtual pallet is sent to the check on *TH* (step 6 of the proposed methodology).

The complete list of combinations generated by the second solver in the static approach is therefore replaced by a dynamic list that is updated at each step.

4.1.3. Dynamic approach with threshold

A significant boost of quest speed is also offered searching a good solution instead of the best solution (bottom part of Figure 5).

As in the dynamic approach, the third solver proceeds by sequential steps.

At each step *nb* virtual pallets are generated, a check on the constraints is performed (MATCH CRITERIA in figure 6) on the *nb* generated combinations: all the virtual pallets that do not satisfy the constraints described in section 2 are deleted. The first of the remaining combinations is sent to the check on *TH* (step 6 of the proposed methodology).

The threshold solver, allocating in the memory only the current n_b combinations stresses less the computer resources. The dynamic approach with threshold finds good results fast: the best solution is not searched any more.

5. NUMERICAL RESULTS

The solver is written, for portability reasons, on Microsoft Excel 2003 and Visual Basic. The performance of the solver, described in sub-section 3.1.3 has been tested on a workstation IBM Intellistation Z Pro (4 CPU: Intel(R) Xeon(R) 5160 @ 3.00GHz, 16,0 GB RAM). The results are reported in Table 1.

Table 1: Results relative to the threshold configuration

Moulds/ pallet	Overall moulds	Cases analysed	Free surface	Solution s found	Work time
1	100	2.0*10^2	80 %	20	4 s
2	100	4.0*10^4	90 %	12	19 s
3	100	8.0*10^6	36 %	10	46 s
4	100	1.6*10^9	24 %	8	18 min
6	100	6.4*10^13	30 %	15	42 min

Five moulds configurations (1, 2, 3, 4 and 6 moulds/pallet) have been tested. The solver work time increases with the number of the moulds/pallet. The moulds used for the test are relatively small; the 6 moulds/pallet configuration finds better solutions (maximum 30% waste space) than the 1 mould/pallet configuration (maximum 80% waste space). The threshold *TH* has been increased to 80% in order to find some solutions; 80 of the 100 moulds have a base surface smaller than 20% of the area of the pallet (and so are rejected by the solver).

Finally the numeric tool has been used to organize the moulds of a whole factory storage. The 501 moulds of the storage have an height range from 250 mm to 850 mm. The moulds have been sorted by heights in 10 groups: all the height intervals of the groups are selected in order to have about 50 moulds for each group. The average vertical pallet wasted area is 5%. Ten hours of simulations are necessary to position 501 moulds on the pallets of the storage. The average horizontal pallet wasted area is 10%; the geometrical problem allows a 90% of pallet surface. The moulds are disposed on the pallets according to the maximum load limit of the shelves. The configurations, selected by the numeric solver, are graphically displayed by the solver; a different color is assigned to each mould. The graphic output helps to implement easily the solver outputs.

The results are satisfactory. The average height of the pallets from ground has been lowered, enabling a safer moulds management. The use of solver helped to save 51% of space. The electric lifts now follow shorter paths for handling the moulds in the new compact storage. Already 28% of new space created has been used to insert new moulds in the shelves; the 23% space available will be used in the future to host new moulds.

6. CONCLUSIONS

The problem of optimizing the waste space of industrial warehouses has been studied. The numerical approach often implies the comparative evaluation of a very high number of configurations. The heuristic threshold approach introduced gives accurate quality results in a reasonable time.

Each object to store has peculiar characteristics and specific needs: weight, size, inertia, object family, kind of use, frequency of use, maintenance needs, expiration date etc. The rules and constraints, specific to each family of objects, should be considered simultaneously in a multi-objective performance index. The designed and implemented solver can only minimize the waste space. Good sense is necessary to interpret and evaluate the simulator results. Good results are achieved only if the decision process is defined and discussed involving logistic managers, the storekeepers, and the electric lifts drivers. The multi stages numeric solver proposed helped to successfully optimize the space inside an industrial moulds storage; after the optimization, the shelves host 51% more moulds. The same approach can be used to solve several different logistic problems.

REFERENCES

Bai, R., 2005. *An investigation of novel approach for optimising retail shelf space allocation*, PhD thesis, University of Nottingham.

Bellman, R., 1957.*Dynamic Programming*, Princeton University Press.

Chyuan, P., Yi-Chiuan, L., Zih-Ping, H., Chin-Lun, O., 2009.A Space Allocation Algorithm for Minimal Makespan in Space Scheduling Problems. Artificial Intelligence and Computational Intelligence, *Lecture Notes in Computer Science,* Volume 5855.

Glover, F. and Kochenberger, G. A., 2003. *Handbook of Meta-Heuristics*, Kluwer

Hillier, F. S. and Lieberman, G. J., 2005. *Introduction to Operations Research*, McGraw-Hill, 8th ed.

Martello, S. and Toth, P., 1975.An Upper Bound for the Zero-one Knapsack Problem and a Branch and Bound Algorithm, *European Journal of Operational Research*. 1, 169-175

Martello, S. and Toth, P., 1990a.*Knapsack Problems: Algorithms and Computer Implementations*, John Wiley & Sons

Martello, S. and Toth, P., 1990b.Lower Bounds and Reduction Procedures for the Bin Packing Problem, *Discrete Applied Mathematics*. 28, 59-70

Osman, I. H. and Kelly, J. P., 1996.*Meta-Heuristics: Theory and Applications*, Kluwer Academic Publishers

Reeves, C., 1995.*Modern Heuristic Techniques For Combinatorial Problems*, McGraw-Hill

Schepers, J., Wottawa, M., 1996.*Warehouse Optimization: Packing-algorithms for goods with complex geometries* (pp. 125-131).Springer Berlin Heidelberg.

Toth, P., 1980. Dynamic Programming Algorithms for the Zero-one Knapsack Problem, *Computing* 25, 29-45

AUTHORS BIOGRAPHY

Elvezia M. Cepolina, PhD, assistant professor at the University of Pisa. She is professor of Transport Planning at the University of Pisa and of Transport Design at the University of Genoa. She has been research fellow at the University College London, UK.
Elvezia is the person in charge of scientific and technical/technological aspects in the following European projects for the University of Pisa: FURBOT, Freight Urban RoBOTic vehicle (FP7-SST-2011-RTD-1); PICAV, Personal Intelligent City Accessible Vehicle System (FP7-SST-2008-RTD-1). Elvezia has been Overseas Scientist in the project SPIRAL Scientific Pedestrian Interaction Research in an Accessibility Laboratory, International Joint Project - 2004/R3-EU Europe (ESEP), founded by Royal Society, UK.
Elvezia is author of many papers on international journals and international conference proceedings.
Elvezia is a member of the SIDT (Italian Society of Transport Professors).

Francesco Cepolina, PhD, has received four degrees: Ordinary degree of Bachelor of Engineering in Mechanical Engineering (1997, University of Leeds, UK), five years degree in Mechanical Engineering (1999, University of Genova, IT), PhD in Mechanics and Design of Machines (2005, University of Genova, IT) and PhD in Mechanical, Acoustic and Electronic Sciences (2006, University of Paris VI, FR). Francesco has been studying, researching and working in six different countries: Leeds (UK), Hague (NL), Madrid (SP), Paris (FR), California (USA) and Genoa (IT). Francesco is deeply interested in: surgery and service robotics, logistics, integrated design and manufacturing, mechanical design and three-dimensional modelling. Francesco has produced over 40 publications and 9 patents in the field of mechanics and robotics. Francesco is reviewer of the following journals:"IEEE Transactions on biomedical Engineering", "Sensors and Actuators, Elsevier".