AN EXPERT SYSTEM FOR OPTIMIZED CORRIDOR ROUTING IN FLOORPLAN

María Gloria Del Río-Cidoncha

Juan Martínez-Palacios

Álvaro Canivell-García de Paredes

Department of Graphic Engineering, University of Seville, Spain

Antonio González-Rodríguez

Department of Applied Mechanical, University of Castilla-La Mancha, Ciudad Real, Spain

Abstract

This paper presents an integral solution for the automatic design of corridors of any type of floorplan. One key point on the generation of layouts is the interconnection of all different spaces (rooms). This article shows how an expert system can be efficiently used for the creation of space interconnections. Application presented provides a user oriented graphical interface and outputs corridors designs for a given layout. It computes these solutions upon a set of rules for the generation of corridors (abstract knowledge), that are defined by an expert planner, and a user defined problem that describes the layout in which corridors are to be drawn (concrete knowledge). The programming language chosen is CLIPS (C Language Integrated Production System). The system has been successfully tested against a number of different sceneries and compared with other space inter-routing tools. The obtained results showed the efficiency and fastness of the application.

Keywords: Automation; Expert system; Routing.

Resumen

Este estudio presenta una solución integral para la creación automática de pasillos en cualquier tipo de plantas de edificios. Una de las claves en la generación de distribuciones en planta es la interconexión de los diferentes espacios (habitaciones), una vez realizada su partición. Existen infinidad de maneras, aunque unas más óptimas que otras. Este proceso de optimización es habitualmente realizado por el ingeniero o arquitecto, teniendo en cuenta durante el proceso un número indefinido de elementos, desde los deseos de los futuros usuarios, hasta su propia experiencia. Esta idea es la principal motivación del trabajo aquí presentado, que muestra la utilización de un sistema experto para la creación de interconexiones entre espacios. La solución trabaja sobre una serie de reglas definidas para la generación de pasillos por un experto humano (*conocimiento abstracto*), así como una definición del problema, sobre la cual el sistema deberá encontrar el trazado óptimo de pasillos (*conocimiento concreto*). El sistema ha sido implementado, probado satisfactoriamente sobre multitud de escenarios, y comparado con otras herramientas de planificación de conexiones. Sus resultados han evidenciado la eficiencia y rapidez de esta solución, representando una ayuda apreciable para la distribución de pasillos en plantas con gran número de espacios.

Palabras clave: Automatización; Interconexionado de espacios; Sistema experto.

1. Introduction

Floorplan design has been a recurring object of research by architects and engineers over time (Cao et al, 1990; McKendall and Shang, 2006). Varied solutions have derived from these studies, forming the so called Layout Generation Methods (LGM) (Immer, 1950), which are based in a number of different techniques: exact methods, general methods, graphs, simulated annealing, tabu search, fuzzy logic, genetic algorithms, other various artificial intelligence techniques etc. These methods are mainly focused on the search of optimal layouts that define position and size of the spaces. Probably the best-known strategy is that of Muther (Muther, 1961), the author of Systematic Layout Planning (SLP), which has been the most widely, used method until recently. SLP is a common choice for solving facility layout problems, independently of their nature: industrial floorplan, airports, hospitals, offices, shopping malls, etc.

However, few dedicated work has been found on the way these spaces are interconnected, most probably because in the case of industrial space distribution, spaces are open (not delimited by walls), thus interconnection paths are not so relevant. It is in the field of electronics circuits that solutions have been proposed for space interconnection. Existing programs such as Cadence OrCAD (Goody, 2000) or Tango SCH PCB (Fernandez-Meroño,1995) generate connections in order to optimize several factors for routing components on PCB (Printed Circuit Board) and even in VLSI (Very Large Scale Integration) design processes.

Therefore, the application presented in this article, make an original approach to this matter importing unseen techniques in architecture, which have been used in other fields for finding the shortest network that connects a group of randomly distributed elements on space.

Given a floorplan already partitioned into rooms, the number of ways in which corridors can interconnect rooms is almost infinite. Ideally, a functional, flexible and versatile building should have a layout that optimize aspects such as total useable area, space functionality, number of rooms or corridors length. This optimization process is usually carried out by designers, who have to take into consideration an undefined number of variables, ranging from the building users' necessities to their own experience in floorplan. In this case, the number of possible solutions is large and the variables involved numerous, which evidences the advantage of using computerized tools. A computer programme can take into consideration any number of variables and can sort out an enormous number of possibilities.

2. Expert System

Expert Systems (ES) are a consolidated branch of Artificial Intelligence (AI), which was created in the 1960s. The underlying idea behind ESs is to transfer expertise from humans to computers. When prompted by a user, computers use this information to infer conclusions, offering advice. Like a human expert, it can give advice and explain how it reached a specific conclusion. ESs have been proved to be an useful tool for decision support and problem solving in many fields (Liao, 2004).

Expert systems are able to evaluate thousands of alternatives and find the most adequate among them, based on a set of predefined and easily understandable criterions. The expert system presented here computes one locally optimal corridor routing solution for a given layout, using a set of rules (such as if-then) previously defined by expert designers.

2.1 Problem definition

The work scenario defined for this expert system consists of a floorplan with rectangular rooms that occupy completely the layout area. Corridors are to be routed only on the lines

that delimit rooms, and the solution corridor must be in contact with all rooms, being sufficiently short.

The main difficulty of such problem is that the number of possible solutions is very large. Even for a computer, sorting out all possibilities, trying to find an absolute optimal would take very long. For instance, the average time to compute the solution on a layout with 25 rooms would be around 1 million years (algorithms are time-exponential). Therefore, the approach presented here uses a more refined algorithm for the search, which takes only 127 seconds to solve a 25 room's problem. It may not be the absolute minimum length solution, but it is short enough to make it difficult for a planner to find a shorter solution in a reasonable time.

This application is thus presented as a support for designers in the design of floorplan, helping them to cope with the distribution of corridors especially in layout with large number of rooms. In section 2 some work done on the topic is shown, as an introduction to the problem modelling and solution implementation described in section 3. Experimental results are summarized in section 4, while section 5 presents the overall conclusions concerning the application.

3. Methodology

The problem addressed by the application presented in this article is modelled in the following, according to the problem definition shown in section 1. Thereafter the implemented solution is described.

3.1 Problem modelling

In order to create a common framework for the comparison of corridor planning tools, a series of assumptions have been made in reference to the problem:

- Orthogonal lines delimit rooms and floorplan.

- Rooms occupy the layout completely.

- No corridor crosses a room.

- A room is connected by a corridor if the corridor is in touch with the room at any point.

- Corridors are always drawn on the orthogonal lines, which delimit rooms.

- The set of corridors that form the solution must connect all rooms inside the floorplan.

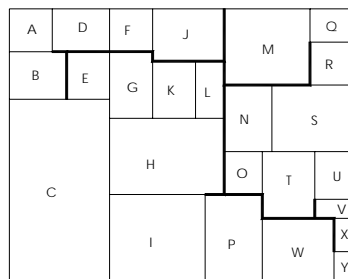Figure 1 shows an example of layout, and the proposed solution, according to these assumptions.

Figure 1. Floorplan with rooms A to Z and the solution corridor

The model is defined based on this floorplan specification, and is formed by the following elements and the relations set among them:

- Arc: each of the four sides that limits a room.

- Node: each point where two or more arcs converge.

- Sub-corridor: a set of arcs that connect a group of rooms.

- Alternative: nodes that can be reached through an arc.

3.2 Solution overview

The solution presented in this paper makes use of different techniques for element interconnection. Generic search algorithms based on artificial intelligence are used together with the fundamental concepts of industrial layout and backed with theories taken from the Steiner problem resolution (Winston, 1994; Darlington, 2000). The main ideas upon which the application works can be summarized in the following:

- The search of an optimal corridor connection among all rooms in a floorplan fits with Steiner's problem definition. However, due to computational constraints, a compromise solution is adopted which finds a local minimum-length solution for the interconnection of all spaces. Thus the chosen strategy is the so-called minimum spanning tree solution. This solution consists of expanding nodes through arcs making the corridor network grow like a tree and always adding the shortest arcs among all alternatives. Nevertheless, a slight adaptation must be made before using this concept. The corridor network must grow no matter the length of the added arcs; the longest arcs must be discarded at the end of the process (taking into account certain restrictions that will be described later).

This corridor expansion could produce loops in the network, as the expansion can be started from more than only one node and does not consider whether nodes are already connected or not.

- Any loop generated during the referred corridor growth could shorten the total length of the corridor; two points on a loop are always connected by two paths, which do not have equal lengths necessarily. Thus all loops must be revised during the final stage of the search, in order to discard the longest paths on them.

- Any corridor search is started by the user choice of compulsory nodes, which are defined as nodes that must be included in the solution corridor, and some other forbidden nodes, that must never be included in the solution corridor.

The architecture of the proposed solution lays on these ideas, which are mainly based on heuristic notions induced by expert designers on floorplan planning. Out of these notions, the following optimal corridor search algorithm has been designed:

1. The user defined compulsory and forbidden nodes are located on to floorplan. Compulsory nodes are the nodes that must belong to the solution corridor (these can be considered starting nodes for the solution), whereas forbidden nodes are the ones, which mustn't belong to the solution corridor. Compulsory nodes could represent accesses from the outside of the floorplan. Forbidden nodes could be used to mark conflictive zones, stairs, courtyards, etc.

2. New compulsory or forbidden nodes are inferred according to the nodes located in step 1). In this way, forbidden and compulsory nodes are extended in the floorplan, creating new sub-corridors, as described in the next step.

3. For each compulsory node that has a single alternative, the node is linked with its single alternative using an arc. These actions will create sub-corridors starting from the compulsory nodes, which will grow until a sub-corridor ends up on a node with more than one alternative. The choice of which alternative to take is not taken at this stage, but in the next step.

4. If all rooms have been successfully connected, the algorithm goes to point 5). But if there are still unconnected rooms, an arc must be eliminated. This arc must not be indispensable, e.g. its elimination must not leave any space unconnected. According to heuristic rules, this non-indispensable arc cannot be a link between two sub-corridors, or contain one alternative of another corridor. Thanks to this arc selection criterion, the new sub-corridor network will be able to grow again once the algorithm is back on step 3), this time avoiding the less favourable arcs. In addition, it will make the network have a dendritic growth (which can result onto presumably shorter corridors than a linear growth), whereas it avoids separating corridors that are already connected. The selection process that picks the arc to eliminate is not immediate at all. An index for each arc is defined as a function of its length; it's belonging to another sub-corridor's alternative and the number of sub-corridors that it is linking. This index is directly proportional to the length of the arc, and inversely proportional to the number of alternatives that it contains and the number of corridors that it links. Thus, the arc that scored the highest index is proposed for its elimination. Next, a recursive process is launched upon this arc candidate in order to find out if it is non-indispensable. If it is not indispensable, it will be tagged as forbidden. If it is indispensable, the arc search process will continue until it finds one that can be eliminated.

5. Algorithm goes back to step 2), unless all rooms have been already connected.

6. The solution corridor has been found, which connects all rooms on the layout. A revision of this solution is launched then, in order to reduce the total length of the corridor. For each loop of the solution, the process tries to eliminate arcs so as to reduce the total length of the corridor without leaving any space unconnected.

This algorithm outputs as a result an optimal corridor that inter-connects all the rooms of the floorplan.

3.3 Implementation of the solution

The algorithm above has been implemented using an expert system platform. An expert system can infer and deduce upon a set of heuristic rules (abstract knowledge) and a set of specific concepts that define the problem scenario (concrete knowledge). The algorithm can be easily encoded as logical rules in the shape of abstract knowledge, whereas a floorplan can be defined as a set of logical parameters, forming the concrete knowledge. This makes the use of an expert system suitable for this case. Furthermore, using an expert system allows a clear and explicit access by expert designers to the set of rules (abstract knowledge), easing the portability and modification of the algorithm.

The complete implementation of the system (an application named NS2) is formed by two different software modules: the user interface and the expert system in it self. The user interface is a front-end for the expert system and is implemented in Borland C++.

The expert system is the core of the application and consists of an inference engine, a knowledge base (the abstract knowledge described above) and a concrete knowledge, defined by each floorplan that is to be interconnected. CLIPS (*C* Language Integrated Production System) has been chosen for the programming of the expert system. There are three main modules in the developed CLIPS software, which are described as following:

- Floorplan pre-processing. This phase loads a layout into memory, creating all elements (nodes, rooms, arcs, etc.) and all the interrelations among them.

- Floorplan processing. This is the main module of the CLIPS architecture, as it computes the optimal corridor on the layout loaded on memory. It executes automatically steps 1) to 5) of the algorithm.

- Solution revision. This stage revises the solution generated by preceding module. It executes step 6) of the algorithm.

All in all, the implemented algorithm is based on artificial intelligence techniques, applied to the generation of trees and connections, using as a reference heuristic rules for floorplan design. As the time required for the computation of solutions of this kind has an exponential tendency, the whole design considers an underlying compromise between processing time and solution optimality, offering to designers useable, efficient and minimally optimal solutions at any time.

As a whole, the implemented algorithm is a piece of software conformed by a total of 7 different logical classes, 3 of them used in the pre-processing phase and 4 of them used in the other two phases. The code has a total of 1007 code lines, and can run as a stand-alone application under windows.

4. Results

The architecture shown in section 3 represents the core of a neatly functional application that allows a non-expert user to input arbitrary floorplan into it, getting solution corridors for them as an output. In the following, a commented example of layout resolution is presented, so as to show how the system works. After that the test bench used to validate the tool is introduced and a summary of the results is shown. Finally, the performance of this application is compared with another program of similar purpose.

4.1 Case study

Once the system loads the layout and the user defined compulsory and forbidden nodes, it infers alternatives, plus new compulsory and forbidden notes.
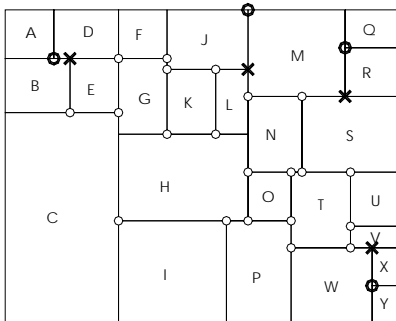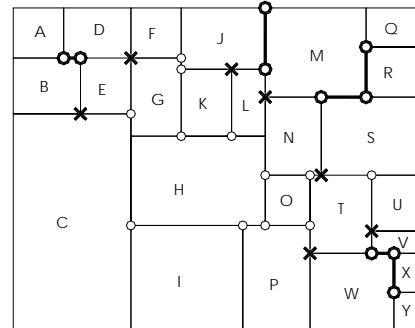

Figure 2. Floorplan for a case study

Figure 3. Floorplan on the first phases

Figure 2 shows the example upon which the behaviour of the algorithm will be described. In the figure, user defined compulsory nodes are marked with an 'O', forbidden nodes are not marked at all, nodes that can be freely used to connect rooms are marked with an 'o', whereas detected alternatives for the compulsory nodes are marked with an 'X'.

As described in section 3, the algorithm creates now corridors from the compulsory nodes to their single alternatives, until no more compulsory nodes have only one alternative. Figure 3 show the results of this network growth stage.

Reached this point, the algorithm tries to find an arc to eliminate, so that once back into the network extension stage shown above, the network can grow again. According to the selection criteria shown in the methodology section, arc number 1 in Figure 4 is eliminated. As this elimination does not leave any sub-corridor with a single alternative, a new arc has to

be eliminated. The second arc chosen is arc number 2. Once arcs 1 and 2 are eliminated, space I has a single alternative, thus the network can grow, creating sub-corridor number 3.
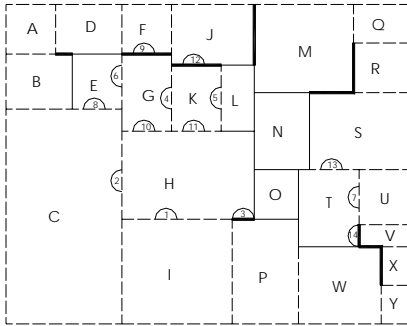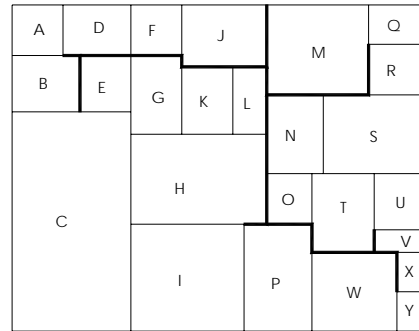

Figure 4. Arc elimination phase

Figure 5. Solution before revision

The process of arcs elimination and network growth goes over and over, until all spaces are finally interconnected. Figure 5 shows the solution corridor found for this floorplan case study. Solution is reached only 16.79 seconds after the algorithm is launched. In this case, the revision of the solution (which took only 7.47 seconds long) proposes no enhancement, as there are no loops in the corridor.

4.2 The test bench

The algorithm has been tested on its velocity and quality of computed solutions. NS2 has been run on a complete battery of layouts, with different number of rooms, total area and morphology. Figure 6 shows the elapsed time of that it took the algorithm to find the solution versus the number of spaces to interconnect.
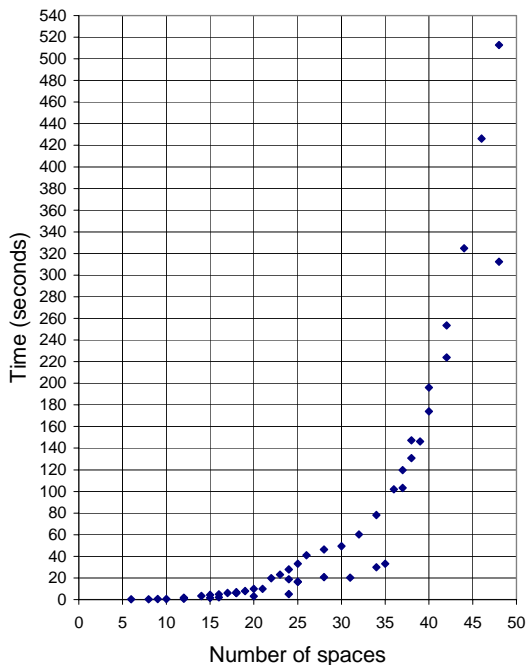

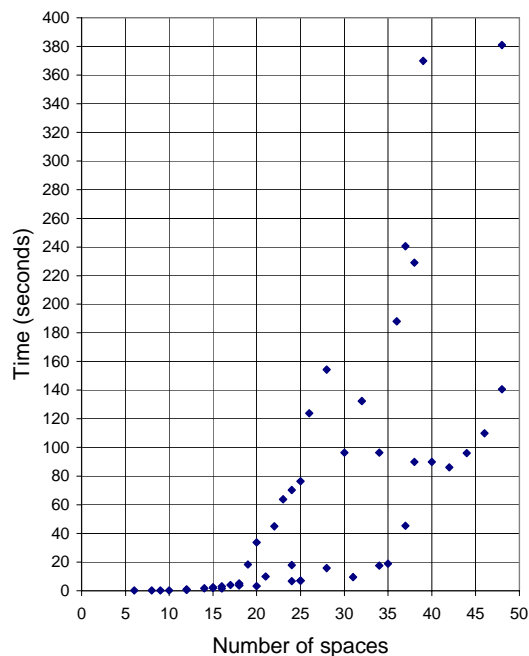Figure 6. For initial solution calculation

Figure 7. For initial solution revision

Figure 7 shows the time it took the system to revise the solution. The elapsed time for the initial solution is exponential with the number of spaces, growing considerably above 35

spaces. This result is coherent, according to the existing exponential-time algorithms for problems of this nature. The high dispersion in time values is related to the way in which cases (floorplan) where randomly generated for the test battery.

4.3 Comparing the application against other approaches

Few are the software approaches applied to space interconnection and route planning. Some applications are found for electronics, particularly for the design of printed circuit boards (PCB). Thus, NS2 has been tested against Orcad, an electronic circuits and PCB design suit. There are other newer programs of this kind, such as Tango, which have been discarded for comparison as not being able to work without the interaction of a person. Furthermore, Orcad offers the user the possibility of defining points where paths must never cross. Thanks to this feature, Orcad and NS2 can work on the same problem definition with exactly the same user-defined constraints (in Orcad, rooms are defined as forbidden zones for tracks, or corridors).

As an example Figure 8 and Figure 9 show the solutions calculated by NS2 and Orcad for the same floorplan definition. NS2 computed the solution in 8.72 seconds, while Orcad did it in 3.12 seconds. Tough NS2 is slower; it computes a better solution, i.e. the length of the corridor calculated by Orcad is 1075 while the length of the corridor calculated by NS2 is 1020.
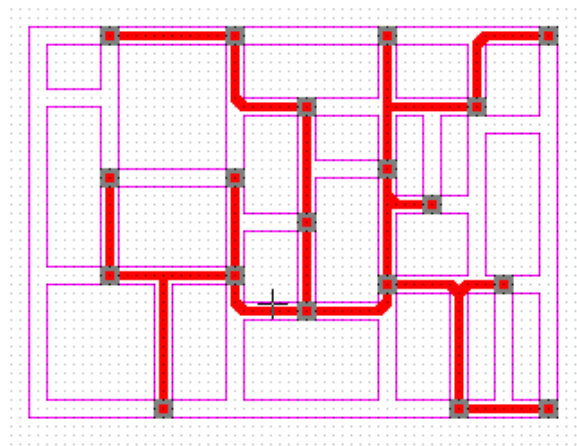

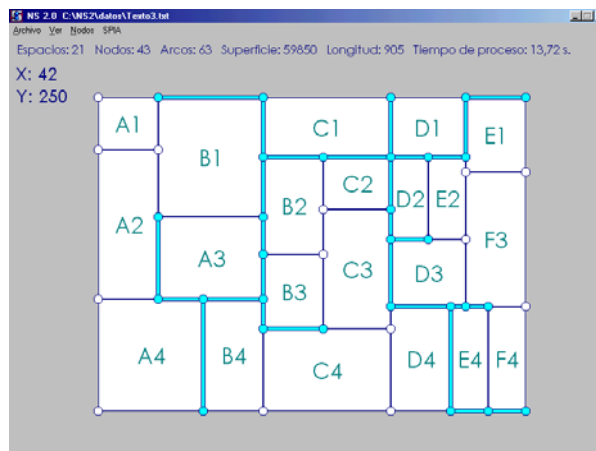Figure 8. Solution found by Orcad

Figure 9. Solution found by NS2

Other tests also confirmed the fact that NS2 is slower than Orcad, but computes better solutions. The time difference is used by NS2 to make its solution shorter than Orcad's.

All these tests show that, though the algorithm may appear to be slow, it is not slow at all, taking into consideration the nature of the problem addressed. In addition, solutions are found without the interaction of any planner, giving the application a desirable autonomy feature, which is not found in other application of this kind. The program is not easily beaten by a person; nevertheless, in some cases a person can enhance a solution generated by NS2. According to this, the application can be used as a stand-alone application or a support application for designers in the design of floorplan.

5. Conclusion

This article presented NS2, a fully functional application that addresses the problem of finding the shortest network that connects elements, which is a problem known by its difficulty. Though NS2's potential applicability is broad, the selected scope in this case is architecture, where space is a floorplan, elements are rooms and the interconnection network is a corridor network. Particularly, for the study case presented here, rooms are chosen to be rectangular and corridors are defined to be routed on the rooms' sides.

An expert system is the engine of the application, suitable for this kind of problems according to its modularity and adaptability. Besides it is easily understandable by designers and provides a good means to translate the way in which designers solve these problems. The result is an inference engine that solves floorplan based upon a set of rules set by designers and concepts directly related to Steiner's problem. Computation time and optimality of the solution have been balanced in order to offer useful solutions in a reasonable elapsed time.

NS2 has been put through an extensive test battery, which has shown the application to be fast enough and offer superior solutions to those found by designers. As usual on this type of problem, the application it is time exponential, though faster than other algorithms. When confronted with other applications, its performance has revealed to be slightly slower than some PCB design applications, though significantly of better quality.

To sum up with, NS2 is a user-oriented application that can be used as a stand alone program or a support design suite for designers on the design of corridor connections in floorplan, being especially useful when the number of rooms is large.

Future improvements include the generalization of the tool, broadening the floorplan model definition exposed in section 3, so as to making the tool applicable on more unusual floorplan. Furthermore, this would make it possible to introduce NS2 in other routing fields such as PCB design, especially once it has been proved to perform better than traditional PCB design tools like Orcad. In addition, a step forward would include its use on path planning for AGVs (Automated Guided Vehicle), offering a value added on the optimization of manufacturing processes.

References

Cao X, He Z, and Pan Y., "Automated design of house-floor layout with distributed planning". *Computer Aided Design* 22, Vol. 4, 1990, pp.213-222

Darlington K., "*The essence of expert systems*". Prentice-Hall, Imprint of Pearson Education, England, 2000.

Fernández-Meroño J M., "*Diseño electrónico por ordenador con Tango SCH PCB Route y PLD*". Paraninfo, Madrid, 1995 (in Spanish).

Goody RW., "*OrCAD PSpice for Windows. Volume 1: DC and AC Circuits*". 3rd ed., Prentice Hall, New Jersey, 2000.

Immer JR., "*Layout planning techniques*", Mc Graw Hill, New York, 1950.

Liao S., "Expert system methodologies and applications, a decade review, from 1995 to 2004". *Expert Systems with Applications*, Vol. 28, 2005, pp. 93–103.

McKendall A R. and Shang J., "Hybrid ant systems for the dynamic facility layout problem", *Computers & Operations Research* 33, Vol. 3, 2006, pp.790–803.

Muther R., "*Systematic Layout Planning*". Industrial Education Institute, Boston, 1961.

Winston PH., "*Inteligencia Artificial*", Addison-Wesley Iberoamericana, Buenos Aires, 1994 (in Spanish).

Corresponding:

María Gloria Del Río-Cidoncha.
Departamento de Ingeniería Gráfica, Universidad de Sevilla, Escuela Técnica Superior de Ingenieros
Av. de los Descubrimientos s/n, Isla de la Cartuja, 41092 Sevilla, Spain
Phone: +34 95 448 61 60
Fax: +34 95 448 61 58
E-mail: cidoncha@esi.us.es
URL: http://www.esi.us.es