7

Multipurpose Layout Planner for Cutting Stock Problems: Implementation Issues

A. Gomes de Alvarenga, Attilio Provedel, F. J. Negreiros Gomes, Hannu Ahonen, L. Lessa Lorenzoni, V. Parada Daza
Universidade Federal do Espírito Santo
Av. Fernando Ferrari, s/n - CEP 29060-900, Vitória-ES - Brasil
email: [agomes, attilio, negreiro, hannu, lessa]@inf.ufes.br
vparada@toqui.usach.cl

H. J. Pinheiro-Pita, L. M. Camarinha-Matos
Universidade Nova de Lisboa
Quinta da Torre, 2825 Monte Caparica, Portugal
email:[hp, cam]@uninova.pt

Abstract

This paper describes a multipurpose cutting stock system that can be used for industrial applications such as glass, wood, shoes, clothing, shipbuilding, etc. We present an architecture based on multi-agent systems consisting of a visualization engine, a KBS, a catalog of algorithms and a dispatching planner for algorithm selection and general control of the system. We also present some examples of application results.

Keywords

Layout Optimization, Interactive Planning, CIM systems, Knowledge-Based Systems, Problem Visualization.

1 INTRODUCTION

There are several kinds of layout problems that appear in the context of manufacturing systems, such as shop floor, textile, furniture and glass industry. In particular, there is a subclass of the cutting stock problem with various applications concerning the objective to minimize the waste of raw material.

The literature presents a lot of approaches and algorithms to solve such problems but there is a lack of Decision Support Systems - DSS to aid the interactive specification, evaluation and solution for them. On the other hand, with respect to the solution approach,

L. M. Camarinha-Matos et al. (eds.), *Balanced Automation Systems II*

existing systems are problem-driven causing the user to lose his working context because of their low degree of flexibility.

This paper presents a balanced automation system for cutting stock problems. The main idea is that the user participates actively in the specification, solution and evaluation of the problem instead of acting only as a data supplier. The user interacts with the system by interchange of examples in the way that the DSS plays the role of a computer-based assistant able to support and amplify the skills of the user faced with complex, unstructured problems.

This paper continues a previous work of Alvarenga et al. (1995) where an architecture for a general layout planner DSS was presented. Here we describe an instance of the cutting stock subclass focusing in the model representation, algorithm selection and user interface. The structure of the paper is as follows: section 2 discusses the cutting stock problems in the industrial view, their application, and algorithms; in section 3, a multiagent object oriented architecture and its compounding modules are presented; finally, section 4 describes examples using rectangular and irregular model representation and their solution presented in a graphic form.

2 THE CUTTING STOCK MODELS, APPLICATIONS AND ALGORITHMS

The cutting stock problem has been widely studied in the last two decades, due to its high applicability in various productive sectors. Hinxman (1980) and Dyckhoff (1990) propose reviews of the accomplished studies, as well as, classifications of the several sorts of problems of this family. A particular case considers cutting of rectangular pieces from an also rectangular plate, satisfying a known demand determined by periodical requirements of clients. Furthermore, all cuts must be of guillotine type, that is, cutting orthogonally, from side to side of the plate or a part of it. It is required additionally that the number of times a given piece must be cut from the plate does not surpass a given value. The objective is to cut the plate with a minimum trim-loss. Several studies have emerged recently in this field. Christofides and Withlock (1977) propose an accurate tree search algorithm using the ideas previously proposed in the classical study of Gilmore and Gomory (1966), who solve a problem with similar characteristics. On the other hand, Wang (1983) proposes an incremental development algorithm to generate the layout. Such algorithm is thereinafter enhanced in the papers of Vasko (1989) and Oliveira and Ferreira (1990). In other line of work, Hinxman (1980) and thereinafter Morabito et al. (1992) approach the unconstrained problem, using the methodology of problem reduction, typically used in problem resolution in artificial intelligence, Pearl (1984). Continuing this trend, Parada, Alvarenga and De Diego (1995) and Viswanathan and Bagchi (1993) represent the problem through an and/or graph. Solution methods proposed in both works, are based on scanning the graph with an informed search method. Both studies show that both Wang's method and their subsequent modifications correspond to an uninformed search in a graph. The constrained bi-dimensional cutting stock problem is an NP-hard problem, therefore no polynomial algorithm to determine the optimum solution has been found. Other approaches based on Simulated Annealing and Genetic Algorithms have been proposed by Parada, Munhoz and Alvarenga (1995) and Parada, Sepulveda and

Alvarenga (1995). Both of them consider a representation of a feasible solution by means of a binary tree. Each node stores information associated with the corresponding cut over the plate. A layout can be completely conformed through a searching process on that tree. In the second case, a syntactic and also binary tree is generated for storing a string representing an individual in a population of the genetic algorithm. In addition to the natural constraints of the problem, many applications involve some special allocation restrictions. These constraints can be:

- Some patterns must be placed on a specific location of the plate, e.g., in the case of shoe manufacturing;
- Some patterns must be placed according to required directions as in the case of clothing industry;
- Specific cutting tolerances regarding the dimensions of the pieces and distances between them.

2.1 Application areas

As mentioned above, cutting stock problems arise in a wide range of industries (e.g., in the glass, paper, wood, steel constructions, shipbuilding, shoe manufacturing and textile industries). In this section, we show two relevant applications of this problem. In some cases it is observed that the nature of the operation in the industry or specific constraints do not permit that these problems can be solved by the conventional cutting stock solution procedures.

Clothing Industry
In the clothing industry the problem of generating a cutting pattern by laying out the pieces onto the stock-sheet has traditionally been solved manually by an expert. Although this approach produces good solutions in terms of trim-loss, it can take more time than an automatic process. In addition to time savings, an automated layout system means better integration in the manufacturing process providing output to the computer-controlled cutting process.

As we can observe, depending on the grain and pattern of the fabric, the orientation of the cutting may be fixed or a finite rotation is allowed. Some specific features of the cutting-stock problem in the clothing industry are:

- Irregularity of the shape to be cut;
- One of the stock-sheet dimensions is relatively large;
- A large number of different stock materials;
- The simultaneous cutting of multiple stock-sheet;
- Specific directions for some patterns are allowed or required.

Obviously when we see the overall objective of the enterprise, the waste minimization is only a subsidiary objective. This involves a series of problems covering short planning periods and more complex objectives that incorporate interaction among periods must be considered. So new features about the problem can be considered and an approach based on Alvarenga et al. (1994) may turn out to be a desirable option for treating the integration of a cutting system in the overall production environment. The interrelations

between the major relevant processes in this environment has been analyzed in Farley (1988).

Shoe Industry
Related to the shoe industry, we observe that the strategy of technical modeling is used as a methodology of manufacturing. Based on this approach, the following facilities are available to the manufacturer:

- Generate or make a copy of shoe model.
- Manufacture the moulds.
- Estimate the amount of raw material.
- Generate the layout of moulds on the leather plate.

Moreover, we can observe a high level of raw material waste generally associated to the following aspects:

- Non-automated process of the technical modeling.
- Generation of mould layout based on empirical methods.

In the case of synthetic materials, the stock plates usually have a rectangular shape.

A more complex case occurs when we have a leather plate which is partitioned in subregions based on technical characteristics of the raw material, Alvarenga et al. (1995).

In the following we will describe some points of two algorithms that will be used as elements of the Algorithms Catalog, introduced in section 3.

2.2 Algorithms

A^* Algorithm
The problem representation is based on a state graph that utilizes a production system characterized by a database, states and a set of rules, Nilsson (1980). Related to the type of the cutting, the following elements need to be considered:

Case 1: Irregular Cutting

- The geometric shapes of the pieces are approximately represented by circumscribed polygons. Moreover, it is assumed that the rectangular stock plate is large enough for fulfilling all demands. In some complex instances of the problem, for example, in the leather cutting, the stock plate is approximately represented as a region delimited by a polygonal boundary.
- The overlap of the pieces is not accepted and rotations of the pieces in the layout are allowed. In some situations, the positions of particular pieces in the layout are restricted to specific plate regions.

Case 2: Regular Non-Guillotine Cutting

- The geometric shapes of the pieces are represented by rectangles.
- The overlap of the pieces is not accepted. Rotation of pieces is permitted.

Details of the representation and a description of the algorithm can be found in Alvarenga et al. (1994). Figure 1 depicts a generic state of the production system.

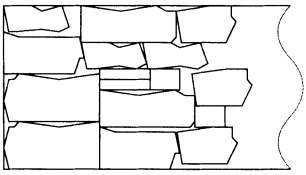

Figure 1 Generic state of the production system.

Simulated Annealing

An approach based on Simulated Annealing has been developed to treat the cutting stock problem. It follows the lines of the presentation given in Lutfiyya (1992). The application of the algorithm based on this technique requires, basically, the establishment of some relevant aspects:

- Solution space.
- Cost function.
- Set of movements that can be used to modify a current solution.

In the conception of the method, there is essentially an analogy with statistical mechanics. Simulated Annealing algorithm can be considered as a generalization of a local search method in the sense that it tries to avoid halting in a local optimum solution. In this way, an analogy between the cooling process of the Simulated Annealing and the search of an optimum solution in an optimization problem can be established. Specifically, the system energy function corresponds to the evaluation function of the optimization problem, a state corresponds to a feasible solution and the lowest-energy state corresponds to the optimum solution of the optimization problem.

The adaptation of the Simulated Annealing algorithm to solve irregular cutting problems, developed in Provedel (1995), was made through the establishment of the following items:

- The initial solution for the searching process is created by a constructive nesting procedure with a random sequence of pieces as an input.
- Initially, the temperature parameter is set to a value high enough. This provides the algorithm a high level of solution acceptance.
- For each solution, the objective function is defined as the waste associated to the current layout.
- The generation of neighbour solution is based on a random procedure, where the algorithm chooses a piece P of the layout and generates a new nesting queue as follows:
  1. In the current nesting queue, the positions of the pieces placed before P is maintained.
  2. The pieces placed after P, including P, are rearranged in a new sequence.
- The temperature is updated based on a geometric cooling.
- A number of iterations in the inner loop of the algorithm is used to describe the thermal equilibrium of the system.

3 MULTIPURPOSE CUTTING SYSTEM

3.1 General architecture

The main objective in the architectural design of the Multipurpose Layout Planner is to construct a modular and generic problem solving system for several layout problems of different types. This is achieved through the principles of agent and object-based programming. The term modularity here refers to a strict division and separation of different problem solving tasks in order to help a programmer and designer to focus his or her decisions on their proper places. The concept of generic problem solving indicates that the same problem solving approaches will be applied in different problem contexts. This makes it possible to select an appropriate problem solving approach with an efficient algorithm without having to rely on some ready-to-use programs written in advance for each possible concrete case.

The basic task in the architectural design of the present system consists of the formulation of the target problems in the way that generic problem solving approaches and their algorithms can be applied to their solution. In the case of the cutting stock problem, this implies that things like the area to be cut, the pieces to be produced, and the criteria of the quality of the cutting must be represented in a way common to all problems.

Similarly, the problem solving approaches and their algorithms have to be written so that they can exploit these common representations. The basic components in the architecture of the Multipurpose Layout Planner consist of agents. Each agent is defined to accomplish a specific subtask in the process of solving the layout problems. Agents in this implementation are mostly units for organizing activities in contrast to autonomous and perceiving agents discussed in the research of Distributed Artificial Intelligence, Huhns (1987).

The principal agents in the system are Model Constructor, Problem Formulator, Algorithm Selector, Process Builder, Algorithm Processor. Additional agents are defined for process supervision, for evaluation of the solution quality and for explanation of the solution process. These are quite similar to those discussed in Alvarenga et al. (1995).

A Model Constructor takes care of given parts of the problem specification, which are common to all problem solving approaches and to all selectable algorithms. It consults the knowledge bases describing the problem setup, i.e., which are the shape and dimensions of the area to be cut, which are the shapes and dimensions of the moulds, and what special information is attached to the cutting process. The Model Constructor is guided by the user via the Visualization Engine. The result produced by the Model Constructor is a set of class instances, which concretely determines the problem specification.

There are three agents, the Problem Formulator, the Algorithm Selector and the Process Builder working together in order to create an instance of the class of Algorithm Processors. The Problem Formulator selects one of the available problem solving approaches (e.g., tree search or simulated annealing), and selects the appropriate components in the problem representation. The selection is, as in the case of the Model Constructor, based on the knowledge base describing the problem, but now the information gathered depends on the selected approach. For example, if the selected approach is the tree search, then

methods for creating child states for a given state as well as methods for evaluating the cost of each state are needed.

The Algorithm Selector determines which of the algorithms available for the selected problem solving approach will be applied to the current problem. It may need to consult the knowledge base of the problem in order to specify algorithm dependent representations needed in the solution process. For example, an application of the A^* algorithm may take advantage of different cost functions than, say, a simple hill-climbing search.

The third agent, the Process Builder, has a more organizational role than the other two. It combines the results of these and produces an instance of a problem solver, an Algorithm Processor. This is an instance of a class of problem solvers, and it contains in its slots information about the problem to be solved and about the selected algorithm.

Both, the problem and the algorithm, are represented by their respective *managers*, which have access to all information on the problem and the algorithm, respectively. It may be worth emphasizing that these managers are created as output of the activities of the three agents used for formulation of the problem, for the selection of the problem solving approach and for the choice of the algorithm inside of it.

3.2 Visualization Engine

Aiming at the systematization of the modeling process and solution of the pieces layout the development of a Graphical User Interface (GUI) is necessary both for the visual representation and for the user interaction.

The Graphical Visualization Engine (GVE) allows the user to describe and visualize his problems in a graphical manner instead of a mathematical notation form used in Bell (1991). The GVE is divided into three modules: Model specification, Example interchange and Solution visualization.

Decomposition of Modules

The design and implementation of the GUI are based in three main paradigms: Visual interactive modeling (VIM), Modeling-by-Example (MbE) and Active Objects Behaviour (AOB), Pinheiro-Pita and Camarinha-Matos (1993).

The VIM approach can be thought of in terms of a more general paradigm called user-centered design which spans the fields of human factors, ergonomics and industrial design, Jones (1994), Bell (1991). This approach supposes that designers should begin, continue and end the design process focused on users' needs.

The MbE approach, unlike the expert systems (where the main objective is to emphasize the autonomous capabilities of the system to solve problems), explores Artificial Intelligence methods, Nilsson (1980), to improve the ability of the system in a high level communication with the user. The goal of the MbE is to support the skills of the decision-maker without replacing his/her judgment. The realization of the MbE approach can be viewed as a two-step process: the first step consists of supplying DSS users with a suitable modeling environment which is, in the second step, enhanced with an inference mechanism whose main objective is to identify problem-solving methods which can be applied to the user-defined models, to activate and run them interactively and communicate their results in a suitable, context-dependent way, Angehrn (1991).

Assuming that the visual aspect is a predominant characteristic in such DSS and that there are different kinds of users, the AOB approach provides support to treat the same

entity in different forms. In this context, depending on the user, the same object could have different views, Camarinha-Matos and Pinheiro-Pita (1993). Figure 2 shows the decomposition of the visualization engine, inside the general Layout Planner Architecture.

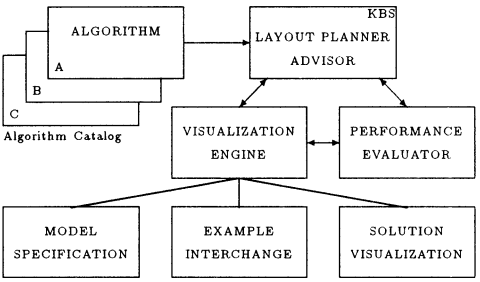

Figure 2 Visualization Engine Modules.

The model specification module allows the user to build an instance of the problem. In this module he/she provides the geometry of pieces and plates, constraints, parameters, etc.

The example interchange module provides a dynamic communication channel used by both the user and the system for the specific purpose of interchanging examples of solution alternatives. It evaluates various instances of produced solutions, suggesting partial solution and new specifications according to its view on the problem context.

As long as the solution is accepted by both the user and the system, the results are exhibited through the solution visualization module.

3.3 Knowledge Base Subsystem

The Knowledge Base Subsystem is composed by the Taxonomy of application problems that organizes the different classes of problems known by the system, and by the Catalog of Algorithms implemented as a library of algorithms.

Each application area is characterized by a specific Domain Knowledge and by the specific Planning Knowledge. The Domain Knowledge contains information about the specific characteristics of the problem, namely, shapes and dimensions of the moulds (or parts to be allocated), typical shape and dimension of the area to be cut, explanations about the various components of the problem, methods to adapt the parts to the user requirements (e.g., a scale method). The Planning Knowledge is composed by sets of rule bases and specifies the constraints and some rules that should be observed for that problem. This knowledge is used by the Model Constructor and Problem Formulator agents.

The Catalog of Algorithms contains, for each algorithm, information about its applicability to a problem, information about specific representation of the problem needed by the algorithm, and an executable code that is used by the agent Process Builder to create an Algorithm Processor.

The Knowledge Base is designed to support maintenance, graphical browsing and

explanation facilities about all of its components.

3.4 Selection of algorithms from the catalog

In the current implementation, written in CLOS (Common Lisp Object System), Steele (1987), the problem solving approaches are represented in the form of classes of problem solvers. Examples of these are the class representing the problem solver in the spirit of the classical Tree Search approach (formulated in a generic manner), and the classes representing the problem solvers using the approaches of Genetic Search, Simulated Annealing and Tabu Search. The algorithms within each problem solving approach are particular implementations of the approach. There are, for example in the case of Genetic Search, several decisions to be made concerning the selection of individuals for the new generation of the population, and there are several variations in the use of the basic Genetic Algorithm.

Each instance of the problem solver class has slots indicating the problem to be solved and the algorithm selected. The items stored in these slots are instances of the classes of problem managers and algorithm managers. They keep track of all details describing the problem and algorithm, respectively. Examples of the former are the description of the area to be cut and the pieces to be produced. The details in the latter case include, for instance, parameters like the number of allowed iterations, the size of the population, and the probabilities of crossover and mutation.

The selection of an algorithm, accomplished by the Algorithm Selector agent, is determined by the information given in the instance of the problem class and in the instance of the problem solving approach (i.e., problem solver). There may be several algorithms applicable to a given problem, as there are several applicable problem solving approaches as well. Thus, it is the task of the user to decide which of the several alternatives should be tried. Different approaches and algorithms can be called and the results obtained can be compared with the help of the Solution Evaluator and Solution Explanator agents.

4 EXAMPLES OF APPLICATION

In this section we present two instances of the cutting stock problem. This illustrates how one of the algorithms in the Algorithm Catalog works.

The following cases were considered:

- Regular Non-Guillotine Cutting
- Irregular Cutting on Uniform Surface

For both cases, the algorithm A^* was used to solve the problems.

4.1 Regular non-guillotine cutting

In the case of regular non-guillotine cutting, a set of 50 pieces (see Table 1) was generated and a sufficient number of rectangular sheets to support the demand of pieces was

supposed. The sheet size considered was 25 × 10.

Table 1 Pieces (length × width) for the test problem.

2 × 1	2 × 1	2 × 1	2 × 2	2 × 2	3 × 3	3 × 3	4 × 1	4 × 1	4 × 1
4 × 2	4 × 2	4 × 3	4 × 3	4 × 4	4 × 4	5 × 1	5 × 1	6 × 2	6 × 2
6 × 5	6 × 5	6 × 5	7 × 1	7 × 5	7 × 5	7 × 7	8 × 1	8 × 1	8 × 1
8 × 4	8 × 4	8 × 4	8 × 6	8 × 6	8 × 8	8 × 8	9 × 2	9 × 2	9 × 4
9 × 4	10 × 1	10 × 3	10 × 3	10 × 7	11 × 2	11 × 2	12 × 3	12 × 3	12 × 3

Figure 3 shows the layout obtained by the algorithm. The remainder pieces are shown in Figure 4.

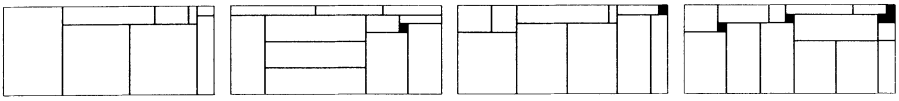

Figure 3 Arrangement of rectangular pieces onto rectangular sheets.

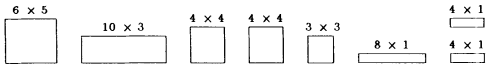

Figure 4 Remainder pieces.

4.2 Irregular cutting

In the case of irregular cutting, a real example (jeans pieces) was treated. Referring to this problem, a demand of 8 pieces (see Figure 5a) with 15 units of each one was arranged on a uniform surface of fixed width and unlimited length (see Figure 5b). Moreover, we supposed that the pieces could be only rotated by 180 degrees from its original orientation.

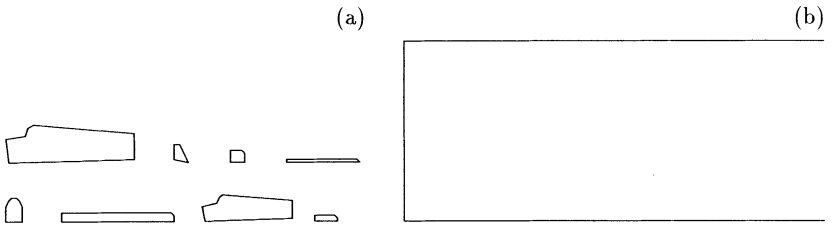

Figure 5 (a) Demand of pieces (irregular cutting) and (b) rectangular stock plate.

Figure 6 shows the layout obtained by the algorithm.

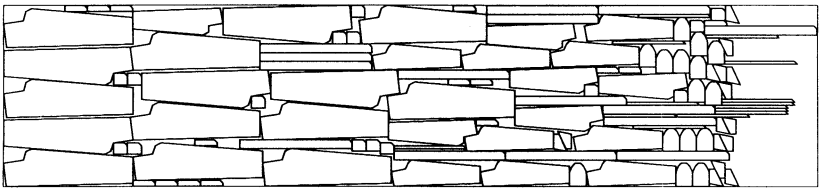

Figure 6 Arrangement of irregular pieces onto rectangular stock plate.

5 CONCLUSIONS

In the sequence of previous articles of the same authors, new developments in the architecture of the Multipurpose Layout Planning were shown. A more detailed description of some investigated algorithms, as well as, of the achieved results of their application on the cutting stock problem were presented. The problem of the multilayer shop floor layout planning and the use of other algorithms are currently under investigation. This problem represents an area where a balanced combination of human decision making and automated solution is a good approach.

6 ACKNOWLEDGMENTS

The authors thank the European Commission in reference to the CIMIS.net project.

The Brazilian authors also thank CNPq (Brazilian Council of Research and Development), through the ProMet project of the ProTeM - phase II, which partially supports this research, and the Portuguese authors would like to thank JNICT for supporting the projects where these ideas were developed.

7 REFERENCES

Alvarenga, A., Provedel, A., Negreiros, F., Parada, V., Sastrón, F. and Arnalte, S. (1994) Integration of an irregular cutting system into CIM. Part I – Information flows, *Studies in Informatics and Control*, **3**, Nos. 2-3, 157-163.

Alvarenga, A., Negreiros, F., Ahonen, H., Pinheiro-Pita, H.J. and Camarinha-Matos, L.M. (1995) Multipurpose Layout Planner, *Balanced Automation Systems – Architectures and design methods*, IFIP, Chapman & Hall, 222-229.

Angehrn, A. (1991) Modeling-by-Example: A link between users, models and methods in DSS, *European Journal of Operational Research*, **55**, 296-308.

Bell, P.C. (1991) Visual interactive modelling: the past, the present, and the prospects, *European Journal of Operational Research*, **54**, 357-362.

Christofides, N. and Whitlock, C. (1977) An algorithm for two-dimensional cutting problems, Operations Research, **25**, 30-44.

Dyckhoff, H. (1990) A typology of cutting and packing problems, *European Journal of Operational Research*, **44**, 145-159.

Farley, A.A. (1988) Mathematical programming models for cutting-stock problems in the clothing industry, *Journal of the Operational Research Society*, **39**, No. 1, 41-53.

Gilmore, P.C. and Gomory, R.E. (1967) The theory and computation of knapsack functions, *Operations Research*, **15**, 1045-1074.

Hinxman, A.I. (1976) Problem reduction and the two-dimensional trim-loss problem, *Artificial Intelligence and Simulation*, Summer Conference, Univ. of Edinburgh, 158-165.

Hinxman, A.I. (1980) The trim loss and assortment problems: A survey, *European Journal of Operational Research*, **5**, 8-18.

Huhns, M.N. (1987) Distributed Artificial Intelligence, *Morgan Kaufman Publishers, Inc.*, Los Altos, California.

Jones, C.V. (1994) Visualization and Optimization, *ORSA Journal on Computing*, **6**, 221-257.

Lutfiyya, H., McMillin, B., Poshyanonda, P. and Dagli, C. (1992) Composite stock cutting through simulated annealing, *Mathl. Comput. Modelling*, **16**, No. 1, 57-74.

Morabito, R.N., Arenales, M.N. and Arcaro, V.F. (1992) An And-Or-Graph approach for two dimensional cutting problems, *European Journal of Operational Research*, **58**, 263-271.

Nilsson, N.J. (1980) Principles of Artificial Intelligence, *Tioga Publishing Company*, Palo Alto, California.

Oliveira, J.F. and Ferreira, J.S. (1990) An improved version of Wang's algorithm for two-dimensional cutting problems, *European Journal of Operational Research*, **44**, 256-266.

Parada, V., Alvarenga, A. and De Diego, J. (1995) Exact solutions for constrained two-dimensional cutting stock problems, *European Journal of Operational Research*, **84**, 633-644.

Parada, V., Munhoz, R. and Alvarenga, A. (1995) A hybrid genetic algorithm for the two-dimensional guillotine cutting problem, in *Evolutionary Algorithm in Management Applications*, Ed. Volker Nissen, Springer Verlag, 183-196.

Parada, V., Sepulveda, M. and Alvarenga, A. (1995) Solution for the constrained guillotine cutting problem by simulated annealing, Submitted to be published.

Pearl, J. (1984) Heuristics: intelligent search strategies for computer problem solving, *Addison-Wesley Publishing Company*, Reading M.A..

Pinheiro-Pita, H.J. and Camarinha-Matos, L.M. (1993) Comportamento de objectos activos na interface grafica do sistema CIM-CASE, *4as. Jornadas de PPPAC*, Lisboa.

Provedel, A. (1995) Um sistema de otimização de cortes industriais e sua integração no contexto CIM, *Dissertação de Mestrado*, UFES, Vitória (ES), Brasil.

Steele, G. (1990) Common Lisp: The Language, Second Edition, *Digital Press*, Bedford, Massachusetts.

Vasko, F.J. (1989) A computational improvement to Wang's two-dimensional cutting stock algorithm, *Computers and Industrial Engineering*, **16**, 109-115.

Viswanathan, K.V. and Bagchi, A. (1993) Best-first search methods for constrained two-dimensional cutting stock problems, *Operations Research*, **41**, 768-776.

Wang, P. (1983) Two algorithms for constrained two-dimensional cutting stock problems,*Operations Research*, **31**, 573-586.