Constrained Two-Dimensional Non-Guillotine Cutting Problem: an Evolutionary Approach

Beraudo V., Alfonso H., Minetti G., Salto C.
Laboratorio de Investigación en Sistemas Inteligentes (LISI)
Facultad de Ingeniería - Universidad Nacional de La Pampa
Calle 110 Esq. 9 (6360) General Pico – La Pampa – Rep. Argentina
Te. / Fax: (02302) 422780/422372, Int. 6302
E-mail: {beraudov, alfonsoh, minettig, saltoc}@ing.unlpam.edu.ar

Abstract

General cutting problems are concerned with finding the best allocation of a number of items in larger containing regions. These problems can be encountered in numerous areas such as computer science, industrial engineering, logistics, manufacturing, among others. They belong to the family of NP-Complete problems. For cases of high complexity deterministic and exact techniques become inefficient due to the vast number of possible solutions that have to be evaluated. In order to reduce the computational load, heuristic or meta-heuristic algorithms are used. The solution method presented in this paper is meta-heuristic based on an evolutionary approach, being its goal to maximize the total value of cut pieces. For that, a modification of Beasley's representation is adopted and for evaluating solutions three placement heuristic rules are developed. Moreover, the effect that placement rules has on Evolutionary Algorithms performance is tested. Computational results are presented for a number of test problems taken from the literature. The results are very encouraging.

1. Introduction

Cutting problems (CP) are a family of natural combinatorial problems, which can be found in areas such as computer science, industrial engineering, logistics, manufacturing, etc. A typology of CP given by H. Dyckhoff [7, 8] distinguishes among the dimension of objects (1, 2, 3, N), the kind of assignment and the structure of the set of large objects ('material', 'object') and of small objects ('products', 'pieces', 'items'). In two- and three-dimensional problems distinctions between *rectangular* and *irregular* cutting problem (pieces of complex geometric forms) can be made. Rectangular cutting may be *guillotin*e, i.e., the current object is always cut end-to-end parallel to an edge.

The solution approaches to this problem can be divided into deterministic and heuristic methods. Deterministic methods are exact techniques that guarantee the fact of finding the optimal solution for a problem. They are usually based on linear programming or enumeration approaches such as branch-and-bound or dynamic programming. For problems of higher complexity these techniques become inefficient, due to the vast number of possible solutions that have to be evaluated in enumeration based approaches. Since conventional methods fail to produce an optimal solution in reasonable amount of computing time, heuristic approaches have been developed. Unlike exact methods, heuristic techniques do not guarantee to find an optimal solution to a problem. They only produce solutions that are "good enough" or near optimal, but at a reasonable computational cost. Solutions will be within a tolerable range of deviance from the optimal solution.

In this paper the two-dimensional cutting problem is tackled, in particular the constrained two-dimensional non-guillotine cutting problem, where a specific number of rectangular pieces are required to be cut from rectangular objects of material. Each piece has a defined value, so the objective is to maximize the total value of cut pieces. In order to solve this problem, an evolutionary option to obtain an optimal solution is proposed. Also different heuristic placement routines were used to distribute the pieces in the object.

The following section gives a brief description of the characteristics of the two-dimensional cutting problem. Section III mentions the different approaches proposed in the literature to solve the general two-dimensional cutting problem. Meanwhile Section IV describes the constrained two-dimensional non-guillotine cutting problem and the notation adopted in this research. Section V describes the solution representation that is used. The following section details the heuristic placement routines with illustrated examples. Section VII summarizes and extends the results in order to analyze the evolutionary algorithm (EA) proposed in this work. Finally, the conclusions and future works are presented.

2. Two Dimensional Cutting Problem

The objective of cutting and packing problems (C&PP) is the efficient allocation of figures in a containment region without overlapping. Hence, the complexity of problems is strongly related to the geometric shape of the items to be cut. Concerning geometry, two types of shapes can be distinguished: regular shapes, that are described by a few parameters (e.g. rectangles, circles) and irregular shapes including asymmetries and concavities (Figure 1).

Since the geometric properties influence the complexity of the problem and the magnitude of the search space, these properties are the most useful for the C&PP classification. Apart from the spatial dimensions of the object, the geometry of the items is a very important criterion [17].

The shape of containing regions can be of different kinds depending on the application area of the problem (see Figure 2); for example, regular shape in the glass industry or irregular shape in paper or textile industry. In case of using regular shape, a single sheet or multiple sheets can be considered. The sheet can be finite in all directions, or infinite in one direction (that is, a roll). For example, in the paper industry the raw material is available in the form of rolls.

In many manufacturing situations, a solution to the single sheet problem is not adequate. If a manufacturing order is given, it is not satisfactory to produce as many of the parts as possible—it is a hard constraint that *all* of the parts on the order must be produced. The number of sheets required (or, more generally, the total area of the sheets used) becomes an unknown in the problem, and the goal is to find a layout that minimizes the number of sheets. This problem, with this objective, is referred to as the *multiple sheet proble*m.

If multiple sheets are involved, either all of them can have the same shape and dimensions, or each one can be different in shape and size. Irregular sheets are crucial for the leather industry: placing the parts on irregularly shaped sheets, rather than on rectangular sheets. "Irregular" means both convex and non-convex sheets, including sheets with holes (defective portions of the sheet).

3. Solution Approaches

A first approach on one dimension cutting problem based on linear programming was proposed by Gilmore and Gomory. In this model each column of the constraint matrix corresponds to a feasible cutting pattern of a single stock length. The total number of columns is very large in practical instances so that only a subset of columns/variables can be handled explicitly. The continuous relaxation of this model was solved using the revised simplex method and heuristic rounding was applied to obtain feasible solutions. Under conditions of cyclic regular production, the solution of the continuous relaxation is optimum for a long-term period because fractional frequencies can be transferred/stored to the next cycle. Heuristic rounding of continuous solutions produces an integer optimum in most cases.

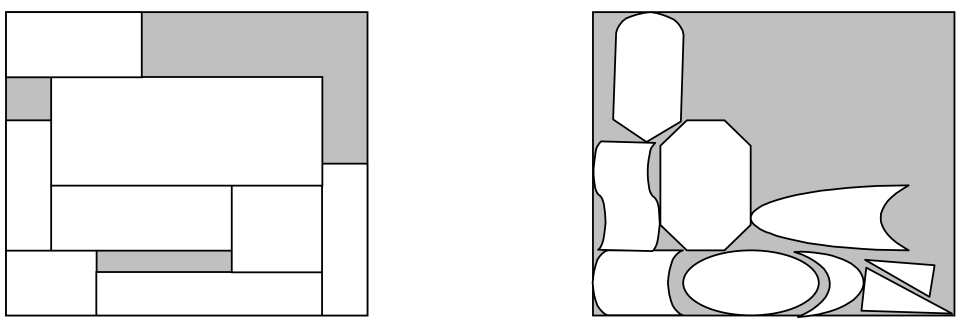

Figure 1. Regular and irregular shapes

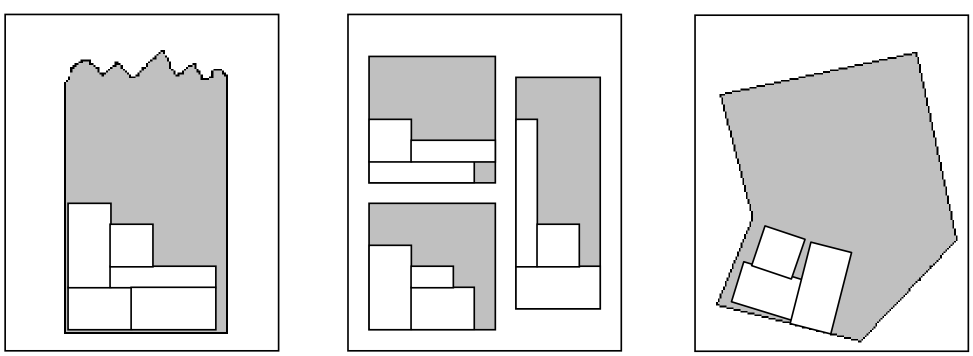

Figure 2. Containing regions shape (rolls, multiple sheets, irregular sheets)

The *unconstrained* two-dimensional *n*-stage cutting problem was introduced in [11, 12], where an exact dynamic programming approach was proposed. The problem has since received growing attention because of its real-world applications. Meanwhile for the *constrained* version, a few approximate and exact approaches are known: [23, 3, 16, 14, 15, 20, 19]. Among exact algorithms, the most successful are linear programming based enumeration without column generation [19] and non-linear programming based enumeration [16].

Combinatorial heuristics are usually called heuristics, which do not use any linear programming relaxation of the problem. Many contributions consider evolutionary algorithms for one-dimensional cutting problem (see [9, 26]) and also for two-dimensional cutting problem [23].

Another group of approaches is based on the structure of the problems. To begin, the various kinds of First-, Next-, and Best-Fit algorithms with worst-case performance analysis are mentioned [25]. The sequential approaches [22, 18, 13] for one-dimensional CP and Wang's combination algorithm [24] and the *extended substrip generation algorithm* (ESGA) of Hifi [15] two-dimensional CP are more sophisticated.

4. Constrained Two-Dimensional Non-Guillotine Cutting Problem

The constrained two-dimensional non-guillotine cutting problem is the one that refers to cutting smaller pieces from a single large planar stock rectangle also called object. Each planar stock rectangle has fixed dimensions (L_0 ,W_0) where L_0 is the length and W_0 is the width, whereas each type of pieces *i* has a length L*i* and a width W_i (i=1,...,m), considering *m* as the number of different types of pieces. Each type of piece *i* has fixed orientation (i.e. cannot be rotated). It must be cut (by infinitely thin cuts) with its edges parallel to the edges of the stock rectangle (i.e. orthogonal cuts); and the number of pieces of each type *i* that are cut must lie between P_i and Q_i ($0 \le P_i \le Q_i$). Each type of piece *i* has an associated value v_i and the objective is to maximize the total value of the cut pieces. It is usual to assume that all dimensions (L_i ,W_i) i=0,1,...,m are integers without entering into a significant loss of generality. To ease the notation in this paper we shall use $M = \sum_{i=1}^{m} Q_i$ [5].

5. Chromosome Representation

The possible solutions to this problem can be encoded inside the chromosome structure. There are many ways to do this, being some better than others. A combined binary/real–numbered solution representation is used, proposed by Beasley [5]. The binary part is:

$$z_{ip} = \begin{cases} 1 \text{ if the } p\text{'th copy } (p=1,\ldots,Q_i) \text{ of type of piece } i \\ \quad \text{is cut from } (L_0, W_0). \\ 0 \text{ otherwise.} \end{cases}$$

while the real-numbered part is detailed as follows. The position of any cut is taken with reference to the centre of the piece,

x_{ip} be the *x*-coordinate of the centre of the *p*'th copy of the type of piece *i*

y_{ip} be the *y*-coordinate of the centre of the *p*'th copy of the type of piece *i*

These centre coordinates are limited by:

$$L_i/2 \le x_{ip} \le L_0 - L_i/2 \qquad i = 1,\ldots, m;\; p = 1,\ldots, Q_i$$
$$W_i/2 \le y_{ip} \le W_0 - W_i/2 \qquad i = 1,\ldots, m;\; p = 1,\ldots, Q_i$$

Unlike Beasley's treatment, in this work real coordinates are present in the chromosome. In order to show an example of this representation, the following problem is considered: m = 3, L_0=10, W_0=10, L_1=3, W_1=7, Q_1=2, L_2=10, W_2=2, Q_2=1, L_3=8, W_3=2, Q_3=2. A possible chromosome conformation is as follows:

Piece (*i*)	1	1	2	3	3
Copy (*p*)	1	2	1	1	2
z_{ip}	0	1	1	0	1
x_{ip}	0	1,5	5	0	4
y_{ip}	0	3,5	8	0	9

In this example the first copy of piece 1 is not cut (z_{11}=0), while the second copy of piece 1 is cut (z_{12}=1) with its centre coordinates at the appropriate real position x_{12}=1,5 and y_{12}=3,5. The other piece to be cut is the only copy available of piece 2 and the second copy of piece 3.

Piece and Copy lists are used as reference to interpret the binary part of all chromosomes. They are maintained unchanged during all the run. The Piece list is built considering all the pieces involved in a particular instance, each piece is numbered and that number is repeated depending on Q_i. The Copy list enumerates the copies for each piece.

Two measures are necessary to evaluate the solutions: V_cutP and V_uncutP; being their difference (V_cutP - V_uncutP) the individual fitness. V_cutP gives the total value of those pieces with Z_{ip}=1 that could be placed in the current layout. Meanwhile V_uncutP is the total value of pieces with Z_{ip}=1 and could not be placed in this same layout; which is used to penalize the individuals and should be next to zero or, at the very least, zero. Taking this into account, the following categorization of the possible solutions came about:

- ✓ First category: this set is formed by solutions with V_uncutP = 0, meaning that all the pieces could be cut.
- ✓ Second category: this set is formed by solutions with V_uncutP < V_cutP.
- ✓ Third category: here the rest of the solutions are placed (V_uncutP > V_cutP).

Here the objective is to maximize the difference between V_cutP and V_uncutP (fitness value); since a solution that contains pieces/copies with z_{ip} =1 but it could not be cut, needs to be penalized and this kind of solution is less desirable than others where all the pieces with z_{ip} =1 are cut. Therefore the best solutions belong to first category.

6. Heuristic Placement Routines

The quality of the layout depends on the sequence in which the rectangles are presented to the heuristic placement routine. The task of the heuristic is to search for a good ordering of the items. A placement routine is then needed to interpret the chromosome and evaluate its quality. Three rules have been developed:

- ✓ **Rule 1.** This rule starts by placing the piece at the bottom left corner of the object. Next the following piece to cut is piled up over the last arranged piece on the left of the object, if it is possible. Otherwise the rule searches for a free space at the bottom left extreme of the object where the piece fits in. One example is shown in the Figure 3 (a) where the piece number four could not be arranged inside the object.
- ✓ **Rule 2.** This is a variant of rule 1, where the priority is to fill the base of the object from left to right instead of piling up. Taking the same configuration from example in rule 1, Figure 3 (b) shows the layout under this rule; in this case piece number five could not be arranged inside the object.
- ✓ **Rule 3.** This rule arranges the pieces in a decreasing way according to their associated values and then rule 1 is applied. The layout made under rule 3, shown in Figure 3 (c), keeps the same configuration than the above examples; in this case all the pieces could be arranged in the object.

7. Experimental Tests And Results

A steady-state evolutionary algorithm, whose objective is to maximize the difference (fitness) between V_cutP and V_uncutP was used. This algorithm implemented: the combined binary/real–numbered solution representation, the uniform crossover and big-creep mutation over the binary part. The created solution should always replace an individual from the population (that was sorted taking the proposed categories into account); the individual to be replaced is selected randomly from the ten worst individuals. To select parents for mating, an adaptated binary tournament selection was used, where the winner belongs to the highest category. In the case two individuals belong to the same category, then one having the highest fitness was chosen.

The algorithms were tested for twelve instances of the Cutting Problem from OR-Library (the problem size and the optimal solution are shown in Table 1) [2, 4]. For each instance, a series of fifteen runs was performed. The maximum number of steps was fixed at 15000 and probabilities for crossover and mutation were set at 0.6 and 0.1, respectively. These values were determined as the best combination of probabilities after many initial trials. The population size was fixed at 120 individuals.

The three different placement heuristic routines were used to evaluate the solutions, generating distinct versions of the algorithm. By preliminary analysis, results of algorithms applying rule 1 and 2 are not being shown; since the best solutions were found by rule 3.

In followings results all instances will be discussed. For each test problem, Table 1 shows: the size of the problem, the optimal solution, the best-found solution, the number of times the optimum was found (#Hits), the average number of steps to obtain the best solution (#Steps) and the average values of mean final populations (Mean F_Pop) are summarized. Optimum solutions are found more than once in instances with m=5 and m=7. For the rest of the instances, values very closed to the optimum are found, maintaining a distance from the optimum lesser than 5.75%. The bigger number of steps needed to find the best value is only obtained in some problems where the chromosome size (M) is big.

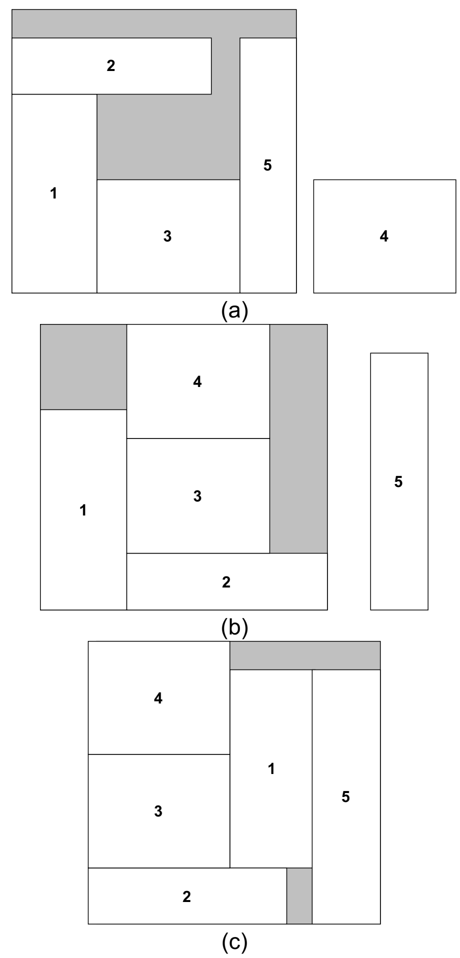

Figure 3. Obtained layout applying the different rules (a) rule 1, (b) rule 2 and (c) rule 3.

Table 1. Results for all instances

Inst	Problem Size (L0,W0)	*m*	M	Optimum	Best solution	#Hits	#Steps	Mean F_Pop
1	(10, 10)	5	10	164	164	29	11.7	104.645
2	(10, 10)	7	17	230	230	3	69.7	178.275
3	(10, 10)	10	21	247	233	0	24.4	148.999
4	(15, 10)	5	7	268	268	3	58.8	146.974
5	(15, 10)	7	14	358	358	2	226.9	222.262
6	(15, 10)	10	15	289	273	0	75.3	179.303
7	(20, 20)	5	8	430	426	0	0.0	219.384
8	(20, 20)	7	13	834	828	0	6.5	557.040
9	(20, 20)	10	18	924	871	0	351.1	559.929
10	(30, 30)	5	13	1452	1383	0	13.8	1004.836
11	(30, 30)	7	15	1688	1688	1	744.5	992.987
12	(30, 30)	10	22	1865	1826	0	1469.7	1231.586

In Figure 4, initial and final population conformations for instances 1, 3 and 5 are shown (this instances are representative of the rest). Each graphic represents the proportion of belonging individuals to the proposed categories. In instances 1 and 5, the individual number that belongs to the first category grew in a considerable way during the evolution. While for instance 3, the most important proportional increment was registered in the second category. In general this evolutionary approach creates more solutions for the two best categories, eliminating solutions where V_uncutP > V_cut and maximizing the sum of v_i values. Averages of these sums are shown in table 1 (Mean F_Pop).

8. Conclusions

In general, the idea for cutting problems is to find the best allocation of a number of items in larger containing regions (objects). As they belong to the family of NP-Complete problems, meta-heuristic algorithms are efficient techniques to solve them. In particular, the solution method proposed here is based on an evolutionary approach for two-dimensional non-guillotine cutting problem considering only regular pieces. In this case, the objective is to maximize the total value of cut pieces. For that, three different placement heuristic rules were devised; being rule 3 the best, which arranges the pieces in a decreasing way according to their associated values and then places the piece at the bottom left corner of the object. Besides, a modified Beasley's representation was developed.

In problem instances with the dimension equal to 5 or 7, the optimal value was found; for the rest of the dimensions, solutions near the optimum were obtained. From analysis of initial and final population conformations, best quality solution arises from evolution, regarding that the majority of the pieces can be cut.

Following steps consist of adding other options to penalize the unfeasible solutions and improving the placement heuristic routines.

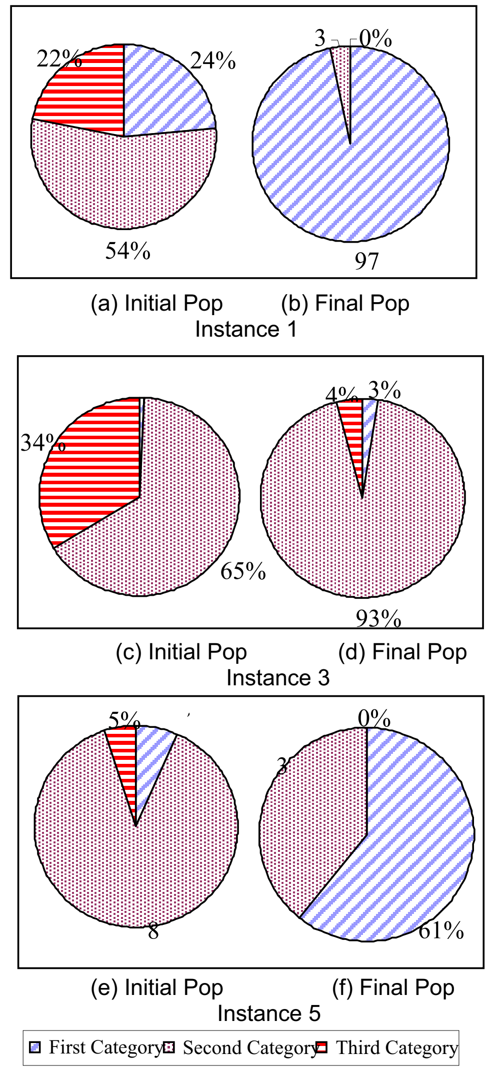

Figure 4: Initial and final populations ordered by the above proposed categorization for different instances

9. Acknowledgement

We acknowledge the cooperation of the project groups for providing new ideas and constructive criticisms. Also to the Universidad Nacional de La Pampa from which we receive continuous support.

10. References

[1] Anand S., McCord C., and Balachander T. "An Integrated Machine Vision Based System for Solving the Non-Convex Cutting Stock Problem Using Genetic Algorithms", University of Cincinnati. *Journal of Manufacturing Systems*, Vol. 18, pp. 396-405, 1999.

[2] Beasley J.E., "An Exact Two-dimensional Non-guillotine Cutting Tree Search Procedure", *Journal of Operations Research* Vol 33, pp. 49-64, 1985.
[3] Beasley J. E., "Algorithms for Unconstrained Two–dimensional Guillotine Cutting", *Journal on Operation Research Society*, Vol 36, pp. 297–306, 1985.
[4] Beasley J.E., "OR-Library: Distributing Test Problems by Electronic Mail", *Journal of the Operational Research Society*, Vol. 41, pp. 1069-1072, 1990.
[5] Beasley J.E., "A Population Heuristic for Constrained Two Dimensional Non Guillotine Cutting". Working paper available from the authors at The Management School, Imperial College, London, England, 2000.
[6] Dasgupta D., and Michalewicz Z., *Evolutionary Algorithms in Engineering Applications*, Springer Verlag, 3º edition, 1996.
[7] H. Dyckhoff, "A Typology of Cutting and Packing Problems", *European Journal of Operational Research*, Vol. 44, pp. 145–160, 1990.
[8] H. Dyckhoff and U. Finke, *Cutting and Packing in Production and Distribution*, Physica Verlag, Heidelberg, 1992.
[9] E. Falkenauer, "A Hybrid Grouping Genetic Algorithm for Bin Packing", *Journal of Heuristics* Vol. 2, pp. 5–30, 1996.
[10] Fritsch A. and Vornberger O. "Cutting Stock by Iterated Matching". Universidad de Osnabruck, Dep. of Math/ Computer Science, Osnabruch. In Operation Research Proceedings, selected papers of the International Conference on OR 94, Springer, pp. 92-97, 1995
[11] Gilmore P. C. and Gomory R. E., "Multistage Cutting Stock Problems of Two and More Dimensions", *Journal on Operation Research*, Vol. 13, pp. 94–120, 1965.
[12] Gilmore P. C. and Gomory R. E., "The Theory and Computation of Knapsack Functions", *Journal on Operation Research*, Vol. 14 , pp 1045–1075, 1966.
[13] Haessler R. W., "Controlling Cutting Pattern Changes in One-dimensional Trim Problems", *Journal on Operations Research*, Vol. 23, pp. 483–493, 1975.
[14] Hifi M., "Exact Algorithms for Large-scale Unconstrained Two and Three Staged Cutting Problems", *Computational Optimization and Applications* Vol. 18, pp. 63–88, 2001.
[15] Hifi M. and R. M'Hallah, "Strip Generation Algorithms for Constrained Two-dimensional Two-staged Cutting Problems", wor ing paper, 2003.
[16] Hifi M. and Roucairol C., "Approximate and Exact Algorithm for Constrained (un)weighted Two-dimensional Two-staged Cutting Stock Problems", *Journal of Combinatorial Optimization*, Vol. 5, pp. 465–494, 2001.
[17] Hopper E., and Turton B. "Application of Genetic Algorithms to Packing Problems - A Review", School of Engineering, University of Wales, Cardiff Newport Road, Cardiff CF2 3TD, UK. In proceedings of the 2º On-line World Conference on Soft Computing in Engeneering Design and Manufacturing, Springer Verlag, London, pp. 279-288, 1997.
[18] Kupke J., *Lösung von ganzzahligen Verschnittproblemen mit Branch-and-Pric*e, Diplomarbeit, Universit at zu Köln, 1998.
[19] Lodi A. and Monaci M., "Integer Linear Programming Models for 2-staged Two-dimensional Knapsack Problems", *Mathematical Programming*, Ser. B Vol. 94, pp. 257–278, 2002.
[20] Morabito R. and Arenales M., "Staged and Constrained Two-dimensional Guillotine Cutting Problems: A New Approach", Tech. report, Notas do ICMSC 126, Universidale de Sao Paulo, Sao Carlos, S.P., Brazil, 1992.
[21] Milenkovic V.; "Rotational Polygon Containment and Minimum Enclosure Using Only Robust 2D Constructions", Department of Mathematics and Computer Science, University of Miami, P.O. Box 249085, Coral Gables, FL 33124, USA. Communicated by J. Rossignac; 1999.
[22] Mukhacheva E. A. and Zalgaller V. A., "Linear Programming for Cutting Problems", *International Journal of Software Engineering and Knowledge Engineering*, Vol. 3, pp. 463–477, 1993.
[23] Parada V., Palma R., Sales D., and Gómes A., "A Comparative Numerical Analysis for the Guillotine Two-dimensional Cutting Problem", *Annals of Operations Research* Vol. 96, pp. 245–254, 2000.
[24] Wang P. Y., "Two Algorithms for Constrained Two-dimensional Cutting Stock Problems", *Journal on Operation. Research*, Vol. 31, pp. 573–586, 1983.
[25] Yue M., "A Simple Proof of the Inequality, for the FFD Bin-packing Algorithm", *Acta Math. App. Sinica* Vol. 7, pp. 321–331, 1991.
[26] Zelle C. and Burkard R. E., "A Local Search Heuristic for the Reel Cutting Problem in Paper Production", SFB Report No. 257, Institute of Mathematics B, Technical University Graz, 2003.