FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 15 (1990) No. 3-4

DECISION SUPPORT SYSTEM FOR CUTTING IRREGULAR SHAPES – IMPLEMENTATION AND EXPERIMENTAL COMPARISON

J. BŁAŻEWICZ*, M. DROZDOWSKI*, B. SONIEWICKI** and R. WALKOWIAK*

Presented by J. Węglarz

In the paper, a decision support system for cutting (or packing) a rectangular sheet of material into pieces of arbitrary shapes, is presented. The system uses two methods (described in the previous paper [2]) which prefer different types of data and the user may decide which one is more suitable for the problem in question. After a brief description of system data files and its manual, some experimental results are presented.

1. INTRODUCTION

This paper contains a description of the decision support system (DSS) for cutting irregular shapes. The basis of this system are two methods described in the previous paper [2], seriously changed and adjusted for solving the problem in question. This system has been implemented on an IBM PC working under DOS operating system. We will not present here a complete treatment of the problem and the methods considered, which may be found in the above paper. However, for the selfcontainment of the present paper, some basic definitions are given below.

Given a set of elements and a sheet of rectangular material with a constant width, find an allocation of elements in the sheet minimizing waste, that is a length of material required. Additionally we have assumed

1. Every element must lay entirely in the stock sheet.
2. No overlap of elements is admitted.

* Technical University of Poznań, Institute of Computing Science, Poznań, Poland.
** Center for Agricultural Development, Poznań, Poland.

3. Filipping the element from "left" to the "right" side (mirror symmetry) is not allowed.

4. Edges of element are either sections of line or sections of a circle. Element can be concave or convex (Holes in elements must be treated separately).

5. Rotations with some step depending on the algorithm, are allowed.

As we already mentioned two modified approaches (described in [2]) have been implemented in the Decision Support System presented.

The Albano-Sapuppo's approach [1] deals with hard irregular cases minimizing waste generated during allocation of every element. Elements are allocated at the left end of the stock sheet and then placed on the right side of the elements already allocated. At the current stage of computations an element minimizing waste is chosen.

Gurel's approach [3] is well defined for elements of similar size and allowing for their clustering into columns. Therefore the smallest elements are eliminated from automatic allocation. This set of small elements can be allocated interactively or with other method. It is also possible to force allocation of all elements.

Thus, the user himself is able to choose the solution or method that fits his needs best. In general we can say that Gurel's method seems to be faster while Albano-Sapuppo's generates solutions with a lower waste. Gurel's algorithm deals with a class of elements well defined to cluster, Albano-Sapuppo's algorithm does not have such a preference,.

In the following Sections we describe how to use the DSS for irregular cutting, data files formats, some operational conditions, and some test results.

2. HOW TO USE DSS ROZKROJ

On the distribution disc there are following files:

ROZKROJ.EXE DSS program;
ROZKROJ.TXT short help file for the ROZKROJ.EXE;
TEST1.IN – TEST10.IN examples, input data files;
TEST1.IN1 – TEST10.IN1, TEST1.IN2 – TEST10.IN2 examples, intermediate solution files.

To start the program simply write ROZKROJ. Please be aware that any data file, solution file or auxiliary file is to be fetched from or written to the current disk and directory. Thus, there should be enough free space.

After the start of program execution the first screen is displayed. Next the main menu screen appears. We have several options here. Any of them can be chosen in the two ways

– pressing the digit key corresponding to the option number in the menu;

– moving lit up bar, up and down with cursor keys we select an option and by pressing ENTER key-confirm a choice.

ESC key breaks program execution at this point. There are the following options in the main menu.

1. Quit the program.
2. Gurel's method.
3. Albano-Sapuppo's method.
4. Display solutions.
5. Program help.

Now, every option will be described.

Option 1. Quit the program – immediately breaks program execution and exits to DOS.

Option 2. Gurel's method – during execution of this part of the program new solutions are computed. First of all we have to give some data to the program. All file names with extension 'IN' in the current directory contain different data sets. They are displayed name after name. In order to choose any file we can do two things:

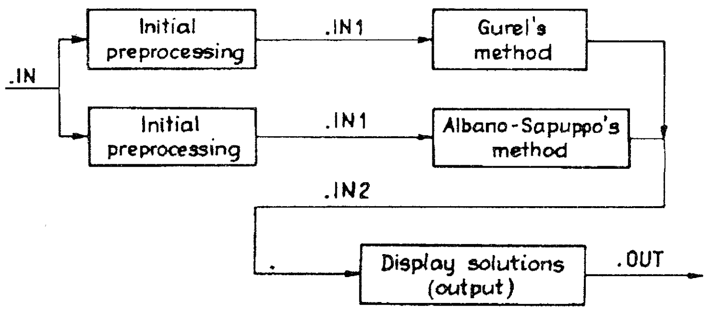

Fig. 1. DSS data file name extensions

– press first letter of the name – be aware that only the first file beginning with this letter is chosen (without any additional confirmation);

– move the lit up bar up, down, left and right to the required file and press ENTER to start computations.

ESC key breaks an execution of this part of program and returns to the main menu.

Data in the file must correspond to the format described in the next subsection (3). Data from the file is preprocessed to find out some kind of errors and to approximate sections of circle with segments of lines. If any error appears, then special message is issued and program breaks the execution. If there was no error new file with extension 'IN1' is created, it is an intermediate data file used by the Gurel algorithm procedure. Then program starts computations. Gurel algorithm routine displays line of twenty

'*' with '---' after the initial and final phase is finished. Allocation of any element in a boundary break or an intermediate break ([2, 3]) is certified by one '*' sign. After filling a certain column of elements, previously issued '*' signs are followed by space and stars again up to twenty characters in the line. An allocation of boundary break one and boundary break two is certified with 'BB1' and 'BB2', respectively. An allocation of an intermediate break is certified by 'IB'. After all parts are allocated, the new solution file is generated with the same name but with automatically added extension 'IN2'. This is a new intermediate format file with elements still in approximated form. Then program abandos Gurel method procedure and starts displaying solution in the graphical form. At this point program switches to the option 4 where current solution is processed.

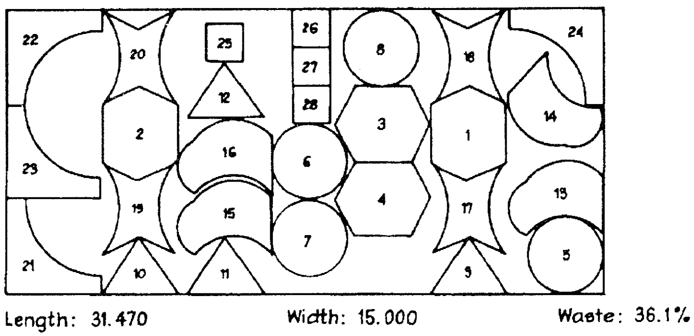

Fig. 2. TEST10 Gurel's method

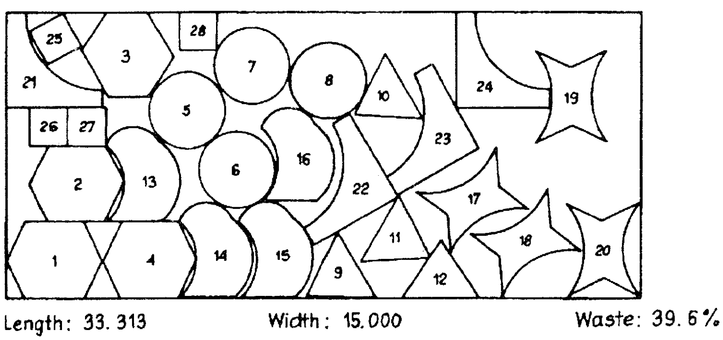

Fig. 3. TEST10 Albano-Sapuppo's method

Option 3. Albano-Sapuppo's method – during execution of this procedure new solutions according to the Albano-Sapuppo's algorithm are computed. The way this routine is handled is identical to Gurel's method described above. The only difference comes from the fact that during computations of a new solution only numbers of allocated elements are displayed.

Option 4. Display solutions – this procedure deals mainly with graphics but not only. In order to display any solution from certain file one has to choose it in the same way input data files have been chosen. Now, all files in the current directory with extension 'IN2' are considered.

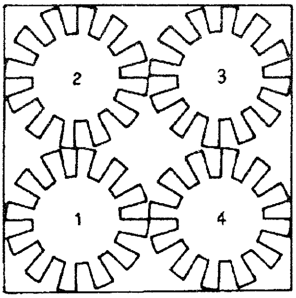

Fig. 4. TEST21 Gurel's method

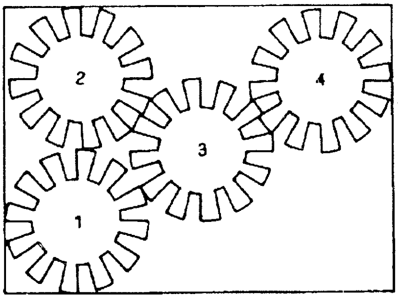

Fig. 5. TEST21 Albano-Sapuppo's method

When solution is displayed we can save it as the final solution or print it. If we save the solution then it is converted from approximated format back to the initial format with sections of circles. Program displays new file name and waits for pressing any key. Then the new file is stored with an 'OUT' extension.

Option 5. Program Help – this routine displays help text file in pages. In the current directory 'ROZKROJ.TXT' file must exist otherwise program issues appropriate message and continues. We can swap pages with PGUP and PGDN keys. ESC returns program to the main menu.

3. DATA FILE FORMAT

The DSS uses files with extension 'IN' as an input, with extension 'OUT' as an output. It generates also two intermediate type files with extensions 'IN1', 'IN2'. Figure 1 explains which module of the DSS utilizes and generates appropriate types of a file.

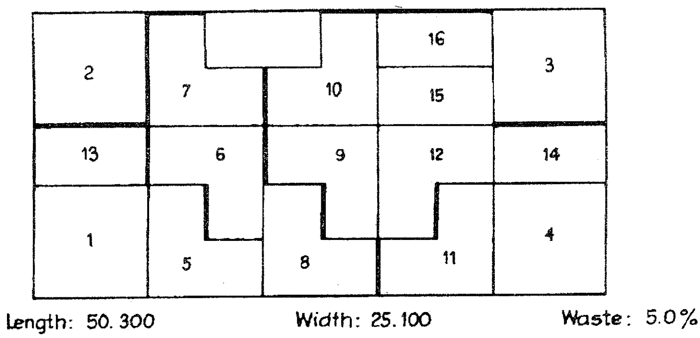

Fig. 6. TEST36 Gurel's method

Now we describe input and output data file format. Let us denote by fl – floating point number with at most ten characters (sign, at most five digits integer part, decimal point, at most three digits fraction), and by int – integer in the range 0 to 32767.

Input file format.

Input file is a text file named with extension 'IN'. It must have a structure as follows

1. Width of the stock sheet (fl).
2. Number of different element shape types (int).
3. Parameters to the algorithms (all fl):
   a) rotation step for the Albano-Sapuppo routine;
   b) percentage parameters for the Gurel's routine;
   c) step length – both routines.
4. Structure of elements. For every type of element shape one has to define:
   a) number of elements of this kind (int);
   b) number of vertices of the shape (int);
   c) description of vertices in the clockwise order (fl)
      – x coordinate;
      – y coordinate;
      – radius of the circle if the current vertex starts a section of a circle as a part of element's edge. If the radius is negative then this

section of circle causes element to be concave (line section from the beginning of circle section to the end is outside the element). Zero if it is a section of line;

– x, y coordinates of the center of that circle, if the edge is a segment of line then components of the normal to this line vector.

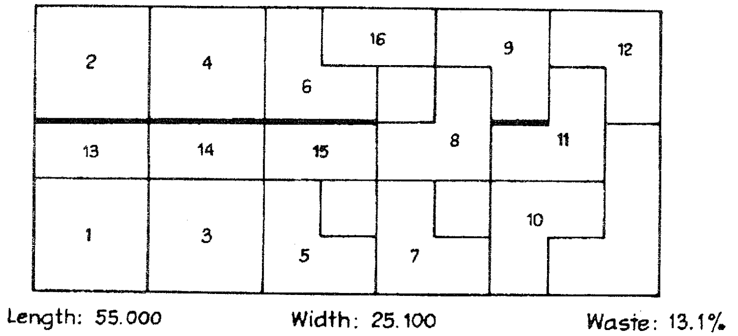

Fig. 7. TEST36 Albano-Sapuppo's method

Every entry of this specification is obligary. Numbers should be separated by space or CR character.

Input file is a text file and can be prepared with any text editor. Specialized program for generation of input data files is described in other paper.

Output file format.

Output file has a name with an 'OUT' extension. It has the following structure

1. Width of the stock sheet (fl).
2. Required length of the stock sheet (fl).
3. Total number of allocated elements of all types (int).
4. Description of elements allocation. Elements in this file are ordered according to the order of different shape types in the input file. For every allocated element
   a) number of vertices (int);
   b) description of vertices in the clockwise order (fl)
      – *x* coordinate;
      – *y* coordinate;
      – radius of the circle if the current vertex starts a section of a circle as a part of element's edge. If the radius is negative then this section of a circle causes element to be concave (line section from the beginning of circle section to the end is outside the element). Zero if it is a section of line;
      – x, y coordinates of the center of circle, if edge is a segment of line then components of the normal to this line vector.

Table 1

Results of automatic element allocation with ROZKROJ ver.2 DSS

Test file name*	Description of element set	Percent. waste	Computat. time	Sheet length
TEST1G	8 convex heksagons	36%	< 1 min	20
TEST1A	”	29.5%	< 1 min	18.149
TEST2G	10 circles	36.3%	1 min	20
TEST2A	”	35.7%	1 min	19.799
TEST3A	10 triangles	37.6%	3 min	9.608
TEST3G	”	45.5%	< 1 min	11.000
TEST4A	10 concave elem. (3 arcs, line)	18.6%	1 min	17.896
TEST4G	”	25.8%	1 min	19.650
TEST5A	12 various elem.	28.2%	2 min	20.595
TEST5G	”	35.3%	1 min	22.870
TEST6G	10 concave elem. 6 vertices, 2 arcs, 4 lines	59.6%	1 min	23.170
TEST6A	”	57.0%	< 1 min	21.784
TEST7A	13 various elem.	35.4%	1 min	19.859
TEST7G	”	37.7%	2 min	20.610
TEST8A	12 various elem.	38.9%	2 min	15.215
TEST8G	”	33.7%	< 1 min	14.000
TEST9A	20 various elem.	36.8%	5 min	27.791
TEST9G	”	38.9%	3 min	28.750
TEST10G	28 various elem.	36.1%	4 min	31.470
TEST10A	”	39.6%	9 min	33.313
TEST11A	10 convex polygons	37.1%	< 1 min	14.583
TEST11G	”	29.5%	1 min	13.000
TEST15A	30 triangles	6.3%	2 min	40.000
TEST15G	”	37.5%	2 min	60.000
TEST24A	4 elem. no arcs normal angles	0.7%	< 1 min	20.000
TEST24G	”	0.7%	< 1 min	20.000
TEST25A	twice elements from test 24A/G	11.7%	< 1 min	45.000
TEST25G	”	3.1%	< 1 min	41.000
TEST26A	4 times elements from test 24	6.5%	1 min	85.000
TEST26G	”	3.1%	3 min	82.000
TEST28A	12 times elements from test 24	2.7%	13 min	245.000
TEST30A	test 25A/G other sheet width	14.7%	< 1 min	35.000
TEST30G	”	25.5%	< 1 min	40.000
TEST31A	twice test 30A/G elements	8.2%	1 min	65.000
TEST31G	”	10.5%	3 min	66.670
TEST35A	test 25A/G other sheet width	20.3%	< 1 min	30.000
TEST35G	”	24.5%	< 1 min	31.670
TEST36A	test 26A/G other sheet width	13.1%	2 min	55.000
TEST36G	”	4.9%	9 min	50.300
TEST38A	12 times elements from test 28A width changed	10.4%	8 min	160.000

* A stands for Albano-Sapuppo method.
G stands for Gurel method.

4. OPERATIONAL CONDITIONS

This DSS can operate on the IBM PC and compatible computers. Suggested minimal hardware configuration is 640 kB of RAM; CGA, HGC or VGA graphic card; floppy disc drive. Optionally hard disc drive and printer. Hard disc device simplifies data operations because all disc operations are executed in the current (where the DSS was started) directory.

5. EXPERIMENTAL RESULTS

Results from several tests are presented in table 1. In that table description of a set of elements, quality of solution (percentage waste), time of computations, are given. Printouts from several example allocations are included in the text.

REFERENCES

[1] A. Albano, G. Sapuppo, Optimal Allocation of Two-Dimensional Shapes Using Heuristic Search Methods, *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-10, No. 5, May 1980.

[2] J. Błażewicz, M. Drozdowski, B. Soniewicki, R. Walkowiak, Two-Dimensional Cutting Problem – Basic Complexity Results and Algorithms for Irregular Shapes, *Foundations of Control Engineering*, vol. 14, No. 4, 1989.

[3] O. Gurel, Circular Graph of Marker Layout, IBM Data Processing Division, *New York Scientific Center Report*, No 320-2965, Feb. 1969.

Received December 12, 1990