Cutting optimization with `piecemaker2d`

Manual for version 3.0.14

Manfred Maennle
`piecemaker2d@maennle.org`

March 30th 2020

1 Introduction

The `piecemaker2d` application calculates cutting instructions that tell you how to cut a (twodimensional) basic plate into the set of pieces you need. `piecemaker2d` is optimizing the cutting plan in order to minimize waste. The tool may be useful for hobby users, craftmen, carpenters, joiners, architects, and others.

Originally, the tool was developed for usage under MS-DOS. Therefore, it is based on command line invocation and file input/output. The tool can easily be called from other applications.

2 Invocation

```
usage: piecemaker2d [options]
options:
  --c=FILE  Use configuration from FILE; do not load a preconfig rc file.
  -C        print copyright- and warranty information
  -h        print this help
  --l=LANG  set language: LANG must be out of ENGLISH, GERMAN
  -V        print version information and compiler settings
```

When being called, `piecemaker2d` does not need any options. Per default, configuration is read from the file `piecemaker2d.cnf` at the current directory. The congif file shall contain all parameters such as input/output files, material name and size of the basic plate, etc. After calculation, the tool writes all results into the output file and, if specified in the config file, calls another application.

If the calculation time gets too long (e. g. for search depth too high), then you can interrupt the run using `Ctrl-C` and restart the calculation with a lower search depth. Be aware, the also the interrupted tool calls a susequent application if defined so in the config file.

2.1 Configuration

Configuraion is in two steps : First step is to read the pre-configuration file `piecemaker2d.rc` which must be located in the same directory as the executable `piecemaker2d.exe`. The pre-configuration specifies the name `CONFIG_FILENAME` of the main configuration file the contains the task-specific settings. The pre-config file may contain some more global settings.

Alternatively to a pre-configuration file, you can use the comman line option `--c=FILE` to specify the main configuration file. In this case, the pre-config file is not read.

The main configuration file contains all task-specific parameters, at least : `INPUT_FILENAME`, `OUTPUT_FILENAME`, `MATERIAL_NAME`, `MATERIAL_LENGTH`, and `MATERIAL_WIDTH`. The main configuration overwrites all prio settings, e. g. from the pre-configuration.

2.2 Format of the configuration file

The configuration file may contain the following settings, see `test\example01` : :

```
/*
 * example configuration file
 *
 */

REM any comment

CONFIG_FILENAME =  <string> (only in pre-config file)
INPUT_FILENAME =  <string> or "(stdin)"
OUTPUT_FILENAME = <string> or "(stdout)"
MAX_TRIALS = <integer> greater than 0, default 3, recommendation max 7
MATERIAL_NAME = <string> name of the basic plate
MATERIAL_LENGTH = <integer> greater than 0
MATERIAL_WIDTH = <integer> greater than 0
CUTTING_N_DIRECTIONS = <integer> 1 oder 2, default 1
CUTTING_WIDTH = <integer> 0 or greater
CUTTING_ALLOWANCE = <integer> 0 or greater
PRINT_SCALE = <integer> greater than 0, default 50
VERBOSE =  <integer> between 0 and 7
FORM_FEED = <integer> 0 or greater
PROGRAM_CALL_ON_EXIT = <string>
```

The `<string>` can be any text. If it contains special characters such as `:`, \, `/`, etc. or spaces, then put the text in between double quotes `"`. `<integer>` means any positive integer number.

Filenames refer to local files or may contain the pathes. The input filename `(stdin)` denominates the standard input device, usually the keyboard. The output filename `(stdout)` indicates the standard output device, usually the current console window.

Search depth mus be greater than `0`, it is recommended to be choosen between `1` and `7` (default is `3`).

The number of cutting directions must be `1`, if the resulting pieces may not be turned by 90 degrees. This is usually necessary for materials that have a directional surface or texture, such as wood grain. A value of `2` means, that pieces may be turned by 90°. If this does not disturb the texture, then choose `2`, since the search algorithm then has more freedom in placing the pieces and calculates better results with less waste.

`PRINT_SCALE` defines the scale factor for the cutting pattern. The larger the values, the larger the print-out. The default value 50 means, the the cutting pattern is printed about 50 characters wide.

The value `FORM_FEED` indicate the number of lines printed per page. The program tries to print cutting patterns as a whole drawing onto prints by adding form feeds between the patterns. The default value `0` means that no form feeds are printed (and, therefore, patterns may be printed on several pages.)

The parameter `VERBOSE` is optional. `0` means no output at all. The higher the value, the more chatty is `piecemaker2d`.

You can define a `PROGRAM_CALL_ON_EXIT` that will be called after terminaten of `piecemaker2d`. If necessary, specify reltive or full paths to the applications. If the string contains special characters or spaces, then put it into double quotes `"`. The call will be executed after any exit of `piecemaker2d`, even after an error or after user interruption.

2.3 Example configuration file

The `bin` directory contains the default pre-configuration `piecemaker2d.rc` that specifies the default name of the main configuration file :

```
CONFIG_FILENAME = "piecemaker2d.cnf"
```

`Example01` in the test directory shows a main configuration example file (`piecemaker2d.cnf`) :

```
/*
 * example configuration file
 *
 */

INPUT_FILENAME = BILL_OF_MATERIAL.TXT
rem OUTPUT_FILENAME = (stdout)
OUTPUT_FILENAME = BEST_SAWING_INSTRUCTIONS.TXT
MAX_TRIALS = 5
MATERIAL_NAME = "WOODBOARD4"
MATERIAL_LENGTH = 2800
MATERIAL_WIDTH = 2050
CUTTING_WIDTH = 4
CUTTING_N_DIRECTIONS = 2
CUTTING_ALLOWANCE = 10
PRINT_SCALE = 50
VERBOSE = 0
rem VERBOSE = 7
FORM_FEED = 0
rem PROGRAM_CALL_ON_EXIT = "dir /a"
```

Following the configuration `piecemaker2d` reads the bill of material from the file `BILL_OF_MATERIAL.TXT` and writes the result into the file `BEST_SAWING_INSTRUCTIONS.TXT`. Sawing instructions are for cutting the basic material `WOODBOARD4` of width 2050 and length 2800 (mm or any other length unit). `CUTTING_ALLOWANCE` adds 10 *mm* additional space to the width and length of each piece. The `CUTTING_WIDTH` defines the width of the saw blade. The program call `dir /a` after termination of `piecemaker2d` is out-commented.

The command line for a program invocation could be for example :
`<path>\bin\win32\piecemaker2d.exe --l=ENGLISH --c=piecemaker2d.cnf`
or just
`<path>\bin\win32\piecemaker2d.exe`

2.4 Input file format

The example input file `BILL_OF_MATERIAL.TXT` containsn the complete bill of materials including names and sizes of each type of piece.

Format :

```
id; item name (not used); material; length; width; number_of_pieces; comment (optional, not used)
```

The input file contains at least colon separated columns. The first column specifies the id of the piece, the third the raw material, then length, width, and the number of needed pieces. The second column can be used to specify a name or the usage of the piece. This column is not used, but must contain at least one character. Each line may contain additional text after the 6th column. It will be ignored from `piecemaker2d`.

The bill of material may contain pice definitions for several materials. `piecemaker2d` uses all lines that match `MATERIAL_NAME` in the third column. All other lines are ignored. If you need cutting instructions for several materials, then simply run the tool several times with the appropriate `MATERIAL_NAME` on the same bill of materials file.

The program does not check the bill of materials for logical errors, e. g. if it contains several lines with the same id (and possibly different sizes). You should use unique `ids` in each line. Otherwise you may have problems the result, because you do not know how to identify the items in the resulting cutting instructions.

2.5 Output file format

The output file contains :

1. A warning, if not all pieces are contained in the resulting cuttung instructions.

2. Complete waste (percentage) and the total number of needed raw material plates.
3. The bill of materials.
4. Data of the configuration files (size and name of raw material plate, parameters)
5. A number of cutting instructions, containing
   (a) Serial number of the instruction,
   (b) statistical information (waste, count of investigated cutting options),
   (c) number of raw plates that mus be cut in this way,
   (d) and a geometrical depiction of the cuts to be performed.

Please check the information of the loaded paramters and bill of material in order to detect possible reading errors of the inpit files.

The geometrical depiction documents in a unambiguous way, how the raw plate must be cut into pices. The depiction shows approximately the proportions of the resulting pieces, denoting each piece name and size. A `CUTTING_ALLOWANCE`, if used, is included in the length and width. Pieces that have the same orientation as given in the bill of materials start with `N` (number). Pieces turned by 90° commence with `Z` (as symbol for a turned `N`) and length and width are flipped accordingly.

Example :

```
overall number of needed raw material pieces: 37

overall waste: 20839466 sqmm = 9.81 %
raw material:
name: "WOODBOARD4", length x: 2800 mm, width y: 2050 mm

cutting allowance: 10 mm
cutting width (of saw blade): 4 mm
cutting directions: 2, i.e. 90 degree turning is possible.
(mamimum search depth for solution search: 5)
(resulting search depth for solution search: 1)
print scale: 50

bill of material from input file:
name: "n10a", length: 2118 mm, width: 0422 mm, amount: 50
name: "n10b", length: 2118 mm, width: 0422 mm, amount: 50
name: "n50a", length: 1910 mm, width: 0422 mm, amount: 11
name: "n50b", length: 1910 mm, width: 0422 mm, amount: 11
name: "n12a", length: 1950 mm, width: 0184 mm, amount: 2
name: "n12b", length: 2000 mm, width: 0184 mm, amount: 7
name: "n12c", length: 1450 mm, width: 0184 mm, amount: 64
name: "n52a", length: 1450 mm, width: 0189 mm, amount: 13
name: "n52b", length: 2000 mm, width: 0189 mm, amount: 1
name: "n13a", length: 1950 mm, width: 0230 mm, amount: 2
name: "n13b", length: 2000 mm, width: 0230 mm, amount: 8
name: "n13c", length: 1450 mm, width: 0230 mm, amount: 77
name: "n14a", length: 1950 mm, width: 0080 mm, amount: 2
name: "n14b", length: 2000 mm, width: 0080 mm, amount: 8
name: "n14c", length: 1350 mm, width: 0080 mm, amount: 77
name: "n15a", length: 0394 mm, width: 0173 mm, amount: 122

cutting instructions no. 1:
(waste: 273664 qmm = 4.77 %; 0000302 search trials)
amount: 4

*--------------------------------------------------------*
|Nn10a                                  |Nn15a   |Zn13b||
|L2128                                  |L0404   |L0240||
|W0432                                  |W0183   |W2010||
|                                       |--------|     ||
|                                       |Nn15a   |     ||
|                                       |L0404   |     ||
|--------------------------------------|W0183   |     ||
|Nn10a                                 |--------|     ||
|L2128                                 |Nn15a   |     ||
|W0432                                 |L0404   |     ||
|                                      |W0183   |     ||
|                                      |--------|     ||
|                                      |Nn15a   |     ||
|--------------------------------------|L0404   |     ||
|Nn10a                                 |W0183   |     ||
|L2128                                 |--------|     ||
|W0432                                 |Nn15a   |     ||
|                                      |L0404   |     ||
|                                      |W0183   |     ||
|                                      |--------|     ||
|--------------------------------------|Nn15a   |     ||
|Nn10a                                 |L0404   |     ||
|L2128                                 |W0183   |     ||
|W0432                                 |--------|     ||
|                                      |Nn15a   |-----||
|                                      |L0404   |     ||
|                                      |W0183   |     ||
|--------------------------------------|--------|     ||
|Nn13b                              |  |Nn15a   |     ||
|L2010                              |  |L0404   |     ||
|W0240                              |  |W0183   |     ||
|-----------------------------------|  |--------|     ||
|                                   |  |Nn15a   |     ||
|                                   |  |L0404   |     ||
|                                   |  |W0183   |     ||
|                                   |  |--------|     ||
|                                   |  |Nn15a   |     ||
|                                   |  |L0404   |     ||
|                                   |  |W0183   |     ||
|                                   |  |--------|     ||
*------------------------------------------------------*

...
```

The example shows the first (from 17) cutting instructions in the directory `example01`. The solid lines depict, whichs cuts must be peformed first, second, and so on. Each field states the name, length and width of the piece, in which a possible cutting allowance is already included. Pieces starting with `Z` are to be turned by 90° (length and width are exchanged accordingly). The `amount` tells you, how many raw material plates are to be cut following the instruction.

All information read (parameters, name and size of raw material, bill of material) are shown ahead of the cutting instructions. Here, the sizes of the bill of material do *not yet* include the cutting allowance.

3 Bug fixing

Program does not start The program cannot start if it does not find the configuration file which is defined by the pre-configuration or by a command line parameter. It then gives an error message.

Program does not terminate Computational effort rises by the number of pieces in the bill of material. If computation takes too much time, then interrupt the program by typing `Ctrl-C` in the command line window. To avoid this, please reduce the parameter `MAX_TRIALS`. A value of 5 is a good choice, it may be reduced down to 1. Please be aware, the a lower parameter `MAX_TRIALS` may yield worse result, i. e. a solution that produces more waste.

Please report errors or write comments and suggestions to my `piecemaker2d@maennle.org` email address.

4 Algorithm

The program `piecemaker2d` implements a two-tiered search algorithm for a solution (cutting direction) that produces the least waste, i. e. that needs the least number of raw material plates.

The inner tier searches a good cutting layout of *a single* raw material plate with respect to the bill of material. The outer tier repeats the inner search loop until all pieces of the bill of materials are covered.

A good cutting layout of an original raw plate is computed by a depth search as follows :
– Determine the biggest piece that fits in the original plate.
– Determine the cutting instruction in order to cut out this piece
– Repeat these steps for all the remaining parts. (I. e., the remaining parts are treated like smaller original raw plates in step one.)

Under some circumstances, the algorithms compares several possibilities and layouts. The number of trials may be limited using the parameter `MAX_TRIALS`. The more trials, the better the result (but cutting layouts may become more complex). As the algorithm performs a depth search, computational effort rises exponentially with the maximum trails per search step.

For practical problems, the algorithm yields very good results at reasonable time of computation. Nevertheless, there is no guarantee the the algorithms finds the optimal solution, even with high `MAX_TRIALS`.

For this example :

```
original [1400x500]
pieces 4
2 [400x400]
2 [300x300]
1 [600x100]
1 [500x200]
```

(with cutting allowance and cutting width equal to 0) the inner optimizsation loop does not fiend the best solution. Obviously, all pieces fit into one original raw plate. The algorithm does not find this solution, because it always uses one piece at a time in order to try the next cut. It should investigate more possibilities, in this case a vertical cut at length 800.

Moreover, even an optimal layout of each single cutting instruction may not yield a globally optimum, as shown in the following example :

```
original [1500x500]
pieces 2
30 [300x500]
30 [200x400]
```

Ten original raw plates are sufficient. But the algorithm finds cutting instructions using eleven plates because it first tries to place the big pieces and afterwards the smaller ones.

Many tests and real usage of the program show that these cases do not often occur in practical usage. To say it in other words : The algorithm may not always find the optimal solution, but it always finds a very good one.