Manual of CompaSS 1.2

Hayward Chan
Department of EECS
The University of Michigan
hhchan@umich.edu

March 18, 2005

Contents

1 Usage **1**

2 Installation **1**

3 File Formats **2**
3.1 Input File Formats 2
3.2 Output File Format 2

4 Basic Options **2**
4.1 Algorithms 2
4.2 Block Orientations 3

5 Advanced Options and Parameters **3**
5.1 Parameters for optimal packing 4
5.2 Parameters for hierarchical packing 4

6 Post-Processing Options **4**

7 Converters and Other Utilities **5**

8 Bug Reports **5**

1 Usage

CompaSS is evoked as follows. The order of the options is not important. If multiple options of the same category appear, however, CompaSS 's behavior is undefined.

```
compass <input-file> <output-file> <input-format> [options and parameters]
```

2 Installation

To compile CompaSS , type `make`. By default, each directory is compiled using `make -j 1`. To alter this, type `make NPROC=x` to support parallel compilations up to x processes.

3 File Formats

File formats `txt`, `soft` and `bbb` are designed to simplify I/O processes while keeping the files human readable. For CompaSS , the extensions of the input and output files are not critical, but we will use extensions `txt` for input and `bbb` for output files to throughout this manual. Converters among `txt`, `bbb`, `soft` and other formats such as `blocks/pl/nets` and YAL format are described in Section 7.

3.1 Input File Formats

`txt` format for hard blocks. The input file is expected to be in either `txt` or `soft` format. Designed for specifying hard blocks, `txt` file describes the dimensions of the n blocks as in Fig.1. For simplicity, no comments is allowed in the first $n+1$ lines of the files, but everything beyond that is ignored. If CompaSS searches for packing where blocks are free to rotate by $\frac{\pi}{2}$, then the order of the dimensions for each block does not matter. The flag `txt` specifies that the input file should be read as `txt` format. Following is an example.

```
compass input.txt output.bbb --txt
```

`soft` format for hard/soft blocks. The input file can also be in `soft` format. It is designed for general packings, which may contain soft blocks. Fig.3 shows a template in `soft` format. The area of ith block is specified by `<area i>`. The aspect ratio of the block can lie between `<a_ratio low_bd_1i>` and `<a_ratio up_bd_1i>`, and similarly up to `<a_ratio low_bd_mi>` and `<a_ratio up_bd_mi>`. Note that the sequence of numbers following `<area i>` must be in non-decreasing order and the aspect ratio lower bound must be strictly larger than the previous aspect ratio upper bound. CompaSS is *not* responsible for any error-checking. In most cases, one interval after the area is sufficient, but the `soft` format offers more flexibility to handle hard/soft blocks with fixed/free orientations. The flag `soft` specifies that the input file should be read as `soft` format. Following is an example.

```
compass input.soft output.bbb --soft
```

3.2 Output File Format

The output file is expected to be in `bbb` format, which describes a packing as in Fig.2. The location of a block refers the location of its lower-left corner. Note that the width and height of a block depends on its orientation in the packing. Therefore, if a block is rotated by $\frac{\pi}{2}$, its width and height will be swapped as they are in the input files.

4 Basic Options

4.1 Algorithms

Small-scale optimal packing. The options `--optimal` or `-o` commands CompaSS to search for an optimal packing (smallest area) of the given set of blocks. The underlying algorithm is a branch-and-bound search. For example, the following sets CompaSS to look for an optimal slicing packing of blocks given by `input.txt` and save it to `output.bbb`:

```
compass input.txt output.bbb --txt -o
```

```
n
<width 1> <height 1>
<width 2> <height 2>
...
<width n> <height n>
```

Figure 1: `txt` format input file template

```
<width of the packing>
<height of the packing>
n
<width 1> <height 1>
<width 2> <height 2>
...
<width n> <height n>
<blank line>
<x-location 1> <y-location 1>
<x-location 2> <y-location 2>
...
<x-location n> <y-location n>
```

Figure 2: `bbb` format output file template

```
n
<area 1> <a_ratio low_bd11> <a_ratio up_bd_11> ... <a_ratio low_bd_m1> <a_ratio up_bd_m1>
<area 2> <a_ratio low_bd12> <a_ratio up_bd_12> ... <a_ratio low_bd_m2> <a_ratio up_bd_m2>
...
<area n> <a_ratio low_bd1n> <a_ratio up_bd_1n> ... <a_ratio low_bd_mn> <a_ratio up_bd_mn>
```

Figure 3: `soft` format output file template

Large-scale hierarchical packing *(default).* The options `--hierarchical` or `-h` commands CompaSS to pack the blocks hierarchically. This does not guarantee optimal solutions but can handle much more blocks (more than 10000). Only hierarchical slicing packing is supported at this moment. The adjustable parameters are described in Section 5.2. The following example evoke CompaSS to look for a sub-optimal slicing packing.

```
compass input.soft output.bbb --soft --hierarchical
```

4.2 Block Orientations

Fixed orientations. If the input format is `txt`, then by default, all the blocks are allowed to rotate by $\frac{\pi}{2}$. All the blocks can be set to have fixed orientations by adding the flag `--fixed`. Note that `--fixed` can only be used with `--txt`, but not `--soft`. Following is an example.

```
compass input.txt output.bbb --txt --fixed -h
```

5 Advanced Options and Parameters

Since CompaSS can handle a great variety of problem instances, it is very difficult to find a default set of parameters suitable for all cases. Therefore, we allow the users to adjust any of the parameters below to optimize the performance of CompaSS . Note that none of the parameters is necessary, and CompaSS can run without specifying any of them. Unless otherwise specified, the parameters are specified as follows:

```
compass input.soft output.bbb --soft --PARAM <VALUE_OF_THE_PARAM>
```

Curve accuracy. In handling soft blocks, we use a piecewise linear curve to approximate the dimensions that can be taken by a block. `--CURVE_ACCURACY` α commands CompaSS to use α line segments to approximate the curve for each aspect ratio range. This parameter is ignored when the input format is `--txt`.

Area accuracy. Setting `--AREA_ACCURACY` to β commands CompaSS to look for the packing within β% from the optimal. It is useful in hierarchical packing when each cluster needn't be optimally packed.

5.1 Parameters for optimal packing

CompaSS looks for an optimal packing by first searching for packing with deadspace less than b%, specified by `--INIT_DEADSPACE`. If not solution is found, it set the deadspace upper bound to be $b\alpha$, where α is specified by `--DEADSPACE_INCRE`. The deadspace is incremented similarly until a solution is found.

5.2 Parameters for hierarchical packing

Under `--hierarchical` mode, CompaSS first groups the blocks into clusters, whose size is specified by `--HIER_CLUSTER_BASE` (s). Each cluster has size at most $s + 1$, and the deviation in total block area among clusters is control by `--HIER_CLUSTER_AREA_DEV`. Within each cluster, similar blocks are considered identical. The degree of similarity accepted by CompaSS is controlled the parameter `--HIER_SIMILARITY_THRESHOLD` (β). For β to be meaningful, it must lie in the range $[0, 2]$, where setting it to 2 means only identical blocks are considered identical, and 0 means all the blocks are considered the same. Setting it to a number greater than 2 switches this feature off.

Similar to branch-and-bound, when CompaSS packs each cluster, it sets a upper bound in deadspace percentage and increases it until a solution is found. The deadspace percentage and increment are controlled by `HIER_INIT_DEADSPACE` (d) and `--HIER_DEADSPACE_INCRE` (γ). At first, the area upper bound is set to be $A(1 + d/100)$. If no solution is found, it increases the area upper bound to $A(1 + d\gamma/100)$, and similarly until a solution is found.

In each cluster, CompaSS looks for optimal solution whose aspect ratio is not extreme. The aspect ratio tolerance is controlled by `--HIER_INIT_AR` (R), `--HIER_WIDTH_INCRE` (δw) and `--HIER_HEIGHT_INCRE` (δh). At first, the width W and height H are limited to $\sqrt{DR}$ where D is the current area upper-bound mentioned above. If no solution is found, the width and height limits are relaxed to $W\delta w$ and $H\delta h$. The limits are relaxed similarly until a solution is found. By default, this aspect ratio constraint is used only when packing clusters in level 0, 1, and 2. This parameter can be specified by `-HIER_USE_AR_LEVEL`. Setting it to a negative value completely switches this feature off. In any case, the aspect ratio constraint is not used when the number of cluster is not greater than s^2 where s is the value of `--CLUSTER_BASE`.

CompaSS tunes the parameters based on characteristics of the input instance, such as the number of blocks, and whether soft blocks are present. User specified parameters are left untouched.

6 Post-Processing Options

All post-processing options are ignored unless CompaSS is in the `--hierarchical` mode.

Compaction. By default, CompaSS compacts the slicing packing into a non-slicing packing, in which no block can be moved left or down with other blocks fixed. This optimization is disabled by the option `--no_compact`.

Controlled Outline. The option `--HIER_OUTLINE_AR` specifies the maximum desired aspect ratio of the final packing. It is followed by a number α at least 1. CompaSS is *not* guaranteed to find a packing with aspect ratio at most α, but tries its best to look for one. The option `--HIER_OUTLINE_DEADSPACE`, together with `--HIER_OUTLINE_AR` further restricts the outline of the packing of interest. `--HIER_OUTLINE_DEADSPACE` is followed by a positive number d, which specifies the dead-space of the outline. In the example below, CompaSS tries to find a packing with at most 4% dead-space and aspect ratio at most 2. Note that having `--HIER_OUTLINE_DEADSPACE` has no effects.

```
compass input.txt output.bbb --txt --HIER_OUTLINE_AR 2 --HIER_OUTLINE_DEADSPACE 4
```

7 Converters and Other Utilities

Since CompaSS needs only the dimensions of the blocks, `txt` and `bbb` formats are designed to simplify I/O processes. The following utilities support useful conversions to or from formats commonly used in VLSI design. More information about the `blocks/pl/nets` and YAL format can be found at `http://www.cse.ucsc.edu/research/surf/GSRC/MCNC/recipe.html`. File converters and other utilities can be found at `http://vlsicad.eecs.umich.edu/BK/FPUtils`.

8 Bug Reports

We gratefully appreciate bug reports or suggestions to make CompaSS a better software. Please contact us at hhchan@umich.edu.