A Parts Arrangement Algorithm for Note PC - An Algorithm for 3D Bin Packing Problem with Arrangement Constraints -

Shigeta Kuninobu†, Keiichi Handa††, Yasushi Sasaki‡, Takashi Iikubo‡‡
† †† Corporate Research & Development Center, Toshiba Corporation
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan
‡ ‡‡ Personal Computer & Network Company, Toshiba Corporation
2-9, Suehiro-Cho, Ome, Tokyo, 198-8710
Tel: † †† +81-44-549-2408 ‡ +81-428-34-2234 ‡‡ +81-428-34-5714
Fax: † †† +81-44-520-1268 ‡ +81-428-30-7457 ‡‡ +81-428-34-7416
E-mail: † shigeta.kuninobu@toshiba.co.jp, †† keiichi.handa@toshiba.co.jp,
‡ yasushi7.sasaki@toshiba.co.jp, ‡‡ takashi.iikubo@toshiba.co.jp

Abstract: We propose a parts arrangement algorithm for equipment such as note PCs. In parts arrangements for actual equipment, a large part of the parts have the arrangement constraints and requirements. The proposed algorithm minimizes the equipment thickness keeping the given equipment depth and width. The algorithm outputs multiple thin arrangement results which satisfy the arrangement constraints and requirements. We evaluated the proposed method by comparing with the results attained by the skilled factory designers.

1. Introduction

In the parts arrangements for equipment such as note PCs, how to attain thin thickness is important which 2D packing problems can not deal with. Baker et al. proposed the first algorithm for a 2D rectangle packing problem [1]. S. Imahori et al. proposed a rectangle packing algorithm to minimize general spatial cost associated with the locations of rectangles [2]. Several 3D algorithms which can deal with the thickness have been proposed [3]. These algorithms allow the parts to be arranged anywhere. In parts arrangements for actual equipment, however, a large part of the parts have the arrangement constraints and requirements. They are, for example, “Arrange the part on the top of the printed-circuit board (PCB)” and “Arrange parts at front as much as possible” respectively.

The proposed parts arrangement algorithm minimizes the equipment thickness keeping the given equipment depth and width, and satisfying the arrangement constraints and requirements.

The following arrangement constraints must be considered in order to output practical arrangement results.

- A rectangular region in x-y plane (horizontal plane) called *set possible region* is defined for all parts. The set possible region defines region where the parts can be arranged (see Figure 1).

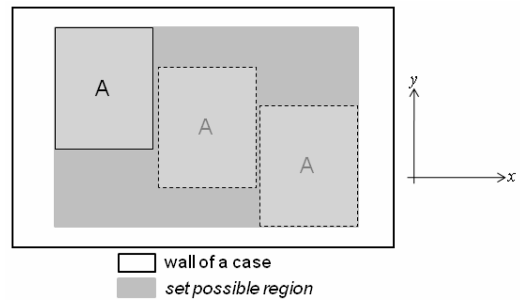

Figure 1 Set possible region of a part A

- Every part belongs to one of the *setting layer*. Five setting layers are prepared in note PCs, case-bottom, board-bottom, board-top, case-top and anywhere. The setting layer “anywhere” is a layer for parts which can be arranged anywhere in the case (see Figure 2).

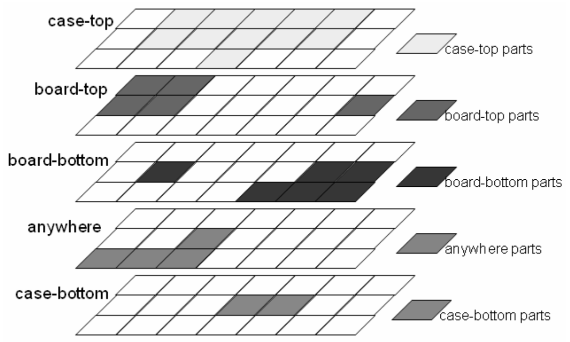

Figure 2 Setting layers

- Some of the parts which belong to the setting layer “anywhere” have to be connected with a *normal connector*. The normal connector belongs to setting layer “board-bottom” or “board-top” (see Figure 3).

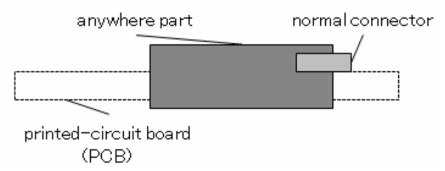

Figure 3 Normal connector

Furthermore, most PC parts have an arrangement requirement such as “Arrange parts at front as much as possible.”

The algorithm explained in section 2 outputs multiple thin arrangement results which satisfy above mentioned arrangement constraints and requirements.

2. Algorithm

Inputs to the algorithm are width and depth of a PC case, a set of parts names and the parts data including parts sizes, setting layers and the set possible region (see Figure 1). Note that, one or more parts correspond to a parts name.

The PC parts are approximated to rectangular parallelepiped.

In order to satisfy the arrangement requirements, the algorithm arranges the parts according to the *packing strategy*. The packing strategy decides the direction of packing. Packing strategy is defined for each setting layer and each part is arranged according to the packing strategy of the layer to which the part belongs. The designer can define different packing strategy to specific parts if the packing strategy of the layer is not suitable. In the algorithm, the following 8 packing strategies can be used, ”Bottom-Left”, “Bottom-Right”, “Left-Bottom”, “Left-Top”, “Top-Left”, “Top-Right”, ”Left-Top” and “Right-Top.” By defining the packing strategy of part *A* as “Bottom-Left” or “Bottom-Right,” part *A* will be arranged at front as much as possible.

Figure 4 shows the flow of the parts arrangement algorithm. The algorithm uses random multi-start local search (MLS) and outputs multiple arrangement results which satisfy the arrangement constraints and requirements mentioned in the previous section. In Figure 4, h_{target} is a target thickness of a PC case, l is a current loop number, l_{max} is a maximum loop number, h is a thickness after parts arrangement (main process) and h_{min} is a thickness of thinnest arrangement result.

The algorithm outputs multiple arrangement results by changing parts to be arranged and by changing the parts arrangement order.

The rest of the paper explains the ‘main process’ shown in Figure 4. The parts arrangement algorithm classifies the parts into setting layers and arranges parts to the corresponding setting layer (x-y plane) according to the packing strategy. The algorithm builds up the parts of each layer to obtain the 3D arrangement result.

2.1. Parts Arrangement in x-y Plane

When arranging parts in x-y plane (setting layer), the algorithm focuses on a rectangular part shadow projected on x-y plane. Let $S(c)$ be the storage to store lines for all rectangular corners $c \in C$ where C={bottom-left(BL), bottom-right(BR), upper-left(UL), upper-right(UR)}. The following 6 steps are performed to arrange a part *A* in x-y plane. The algorithm repeats from Step1 to Step6 until all parts arrangement is completed.

Step1. The algorithm determines the base point b ($\in C$) according to the packing strategies. The relation between the packing strategies and the base points is shown in Table 1.

Step2. The algorithm makes two *region lines* which represent the set possible region for base points, and stores to $S(b)$. Figure 5 shows how to make region lines.

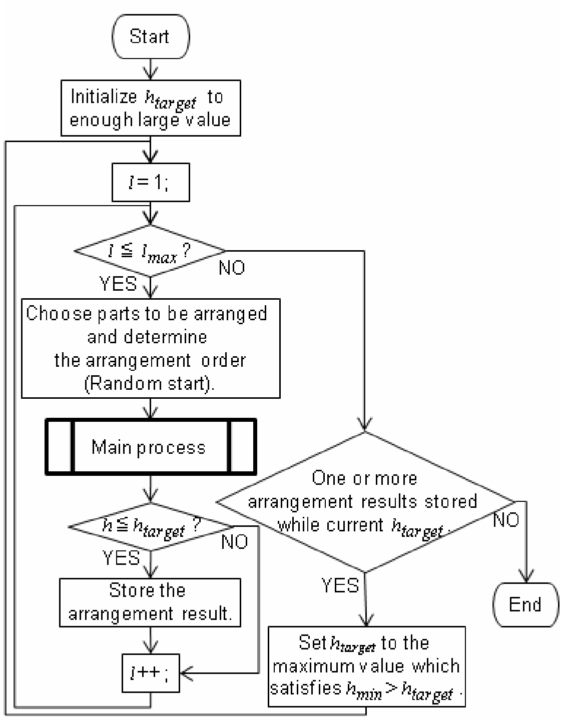
Start
Initialize h_{target} to enough large value
$l = 1;$
$l \leqq l_{max}$?
YES
NO
Choose parts to be arranged and determine the arrangement order (Random start).
Main process
$h \leqq h_{target}$?
YES
NO
Store the arrangement result.
$l{+}{+};$
One or more arrangement results stored while current h_{target}
YES
NO
Set h_{target} to the maximum value which satisfies $h_{min} > h_{target}$.
End

Figure 4 Flow of the algorithm

Table 1 Packing strategy & base point

Packing strategy	Base point
Bottom-Left	bottom-left corner (BL)
Left-Bottom	bottom-left corner (BL)
Bottom-Right	bottom-right corner (BR)
Right-Bottom	bottom-right corner (BR)
Top-Left	upper-left corner (UL)
Left-Top	upper-left corner (UL)
Top-Right	upper-right corner (UR)
Right-Top	upper-right corner (UR)

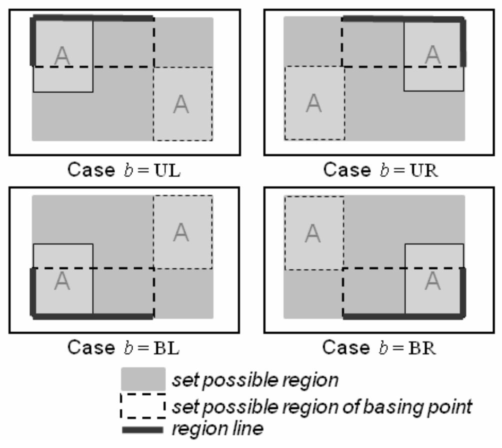

Figure 5 Region lines

Step3. The algorithm finds all intersections of the lines in $S(b)$ by using the plane sweep method and sort intersections based on the packing strategies. The intersections are the candidate points where parts are arranged. The relation between the packing strategies and the sorting methods are shown in Table 2. Only two lines (region lines) are stored in $S(b)$ just before the first part is arranged. However, more lines (*boundary lines*) which are made in Step5 are stored in $S(b)$ when arranging second or later parts.

Table 2 Packing strategy & sorting method

Packing strategy	Sorting method
Bottom-Left	ascending y-coordinate order → ascending x-coordinate order
Left-Bottom	ascending x-coordinate order → ascending y-coordinate order
Bottom-Right	ascending y-coordinate order → descending x-coordinate order
Right-Bottom	descending x-coordinate order→ ascending y-coordinate order
Top-Left	descending y-coordinate order→ ascending x-coordinate order
Left-Top	ascending x-coordinate order → descending y-coordinate order
Top-Right	descending y-coordinate order→ descending x-coordinate order
Right-Top	descending x-coordinate order→ descending y-coordinate order

Step4. The algorithm arranges the base point of A to the intersection found at first which satisfies the following conditions.

- Part A does not overlap with other parts which belong to the same setting layer (Parts which belong to the setting layer "anywhere" are excluded).
- Part A is within the set possible region.
- PC case thickness after part A is arranged is below h_{target}. The case thickness is calculated by parts-building process shown in section 2.2.

Since several arrangement candidate points (intersections) are computed, other candidate points are checked when the PC case thickness becomes thicker than h_{target} at a certain candidate point. If no intersection satisfies these conditions, the algorithm searches for an alternative part which has the same part name. When no alternative part is found or any alternative part does not satisfy these conditions, the main packing process exits as $h=\infty$.

Step5. For all rectangular corners $c \in C$, the algorithm makes two *boundary lines* which represent the boundary of arranged parts, and store to $S(c)$. Figure 6 shows how to make parts boundary lines.

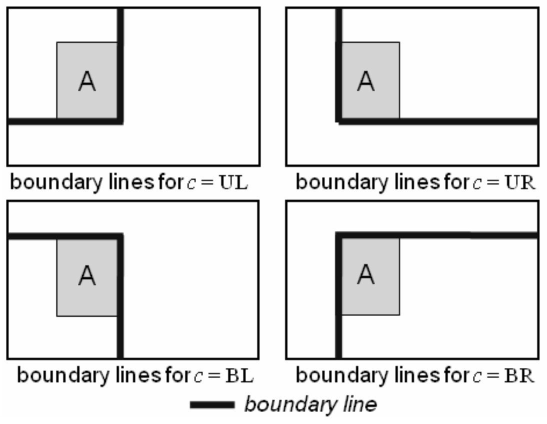
A
boundary lines for c = UL
boundary lines for c = UR
boundary lines for c = BL
boundary lines for c = BR
boundary line

Figure 6 Boundary lines

Step6. The algorithm deletes region lines from $S(b)$ which are made in Step2.

2.2. Parts-building process

The algorithm computes the vertical direction position of each part by building the parts. Parts-building process is executed as follows.

Step1. The algorithm builds the parts which do not belong to the setting layer "anywhere" as shown in Figure 7.

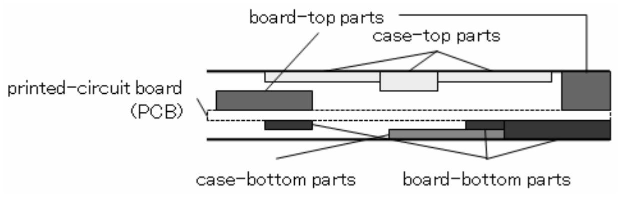

Figure 7 Parts-building

Step2. To arrange part A which belongs to the setting layer "anywhere" and needs normal connecter, the algorithm searches a connectable normal connector which makes the case thinnest. For example, if there are two connectable normal connectors as shown in Figure 8, the algorithm chooses the normal connector 2.

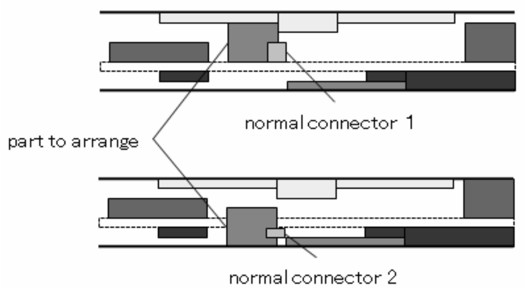

Figure 8 Building a part which needs normal connector

If no normal connector which satisfies the following condition is found, the parts-building process exits as $h=\infty$.

- Part A does not overlap with other parts which belong to setting layer "board-bottom", "board-top" or "anywhere".

Step3. For the parts which belong to the setting layer "anywhere" and do not need normal connecters, the algorithm searches the vertical direction position where make the case thinnest.

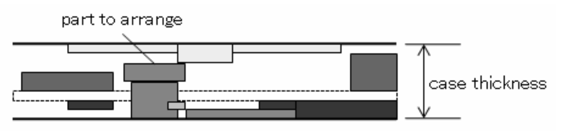

Figure 9 Building a part which does not need normal connector

The main process ends when all parts are arranged in the x-y plane and the parts-building process is finished.

2.3. Other features

The parts arrangement program also satisfies the following arrangement constraints and requirements in order to compute practical arrangement results.

- Arrangement of parts which have mutual position restrictions between them.
- Decentralized arrangement of connectors (e.g. USB connector, IEEE connector, RGB connector...).
- Arrangement of parts which penetrate the PCB (see Figure 10).

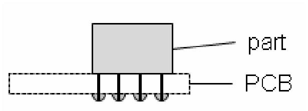

Figure 10 A part which penetrates PCB

- Parts arrangement on the regulation size PCB.
- Parts arrangement with the minimum depth and width while the case thickness is kept below the given target thickness.

3. Evaluation

The parts arrangement program which is implemented by Java [1] output about 20 arrangement results within 2 minutes for mobile note PC whose size is about 30cm×22cm with 39 parts and output about 20 arrangement results within 5 minutes for AV note PC whose size is about 44cm×30cm with 57 parts. In both cases, the results included a similar arrangements attained by the skilled factory designers. The calculation time was short enough to apply the algorithm to actual design process.

Figure 11 and Figure 12 show the arrangement result of AV note PC.

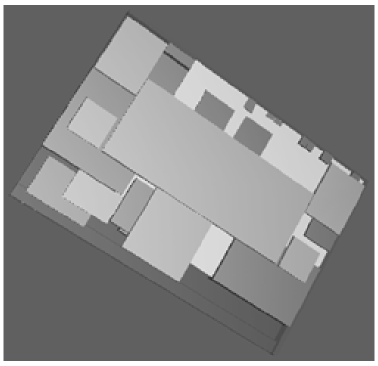

Figure 11 Arrangement result (front view)

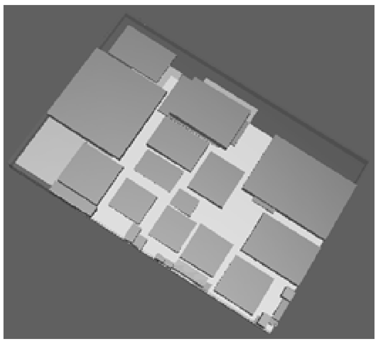

Figure 12 Arrangement result (back view)

4. Conclusion

We proposed a parts arrangement algorithm for note PCs. The algorithm outputs multiple thin arrangement results which satisfy the arrangement constraints and requirements.

We evaluated the proposed method by comparing with the results attained by the skilled factory designers. The proposed method yielded a similar PC arrangement results by the skilled designers. Together with short calculation time, the proposed method is useful enough for actual PC design processes.

References

[1] B.S. Baker, E. G. Coffman Jr. and R. L. Rivest: "Orthogonal packing in two dimensions," SIAM Journal on Computing, 9 (1980), 846-855

[2] S. Imahori, M. Yagiura and T. Ibaraki: "Local Search Algorithms for Rectangle Packing Problem with General Spatial Costs," Mathematical Programming 97 (2003) 543-569

[3] Z. Jin, T. Ito and K. Ohno: "The Three-Dimensional Bin Packing Problem and Its Practical Algorithm," JSME International Journal. Series C, Mechanical systems, machine elements and manufacturing Vol.46 No.1 (2003) 60-66

[1] Java Ver. 1.5.0_11, on Windows XP (Pentium4 2.80GHz, 512MB RAM)