

# BOARD LEVEL PLACEMENT

Jing Lee

Department of Electronic Engineering

Southern Taiwan University of Technology

Tainan, Taiwan 701, R.O.C.

Email: leejing@mail.stut.edu.tw

Because of complexity in modern printed wiring board (PWB) designs which can consist of a large number of components, the placement of the components has become an extremely time consuming task and may involve tradeoffs of equally important weight. To help a design engineer, placement automation techniques have been developed. Most of them are developed for improving wireability [Che84, Got78, Han72, Oda87, Qui79, Sia87] by either minimizing of the total wire lengths, or minimizing the maximum crossings of a horizontal or vertical cut of the placement. More sophisticated program uses a combination of both conditions [Wip82]. Recently, some CAD programs with a minimized system failure rate were also developed. Particularly, Pecht and Naft [Pec87] presented a placement method for reliability based on random sampling of placement configurations. Dancer and Pecht [Dan89] presented several optimization schemes for reliability and compared their accuracy and computational speed. Mayer [May78] discussed the optimization for reliability based on the thermal design constraints of electronic systems in avionics. Osterman [Ost92] discussed component placement for reliability based on a failure model which incorporated component temperature, base operating temperature, threshold temperature, and changes in temperature. Eliasi et al. [Eli90] presented a Monte Carlo thermal optimization method for component placement on a PCB.

It has been observed that both reliability and wireability of a board depend on its module placement. Thus a more general problem is how to place the modules to optimize the two objectives together. In most cases no solution satisfies both conditions simultaneously. Generally, only Pareto optimal solutions exist. The Pareto optimal solution of a two-objective optimization problem is the one for which any further improvement of either one objective function will cause the degrading of the other one. However, Pareto optimal solutions are very difficult to obtain. In previous studies, only Osterman and Pecht [Ost90] presented a novel placement procedure to obtain tradeoff solutions. In their approach, they first constructed a position-adjacency matrix which was mapped from the optimal placement based on reliability. Then, they combined the position-adjacency matrix with the connectivity matrix, constructed from the signal nets by a weighting factor, to form a hybrid matrix. After that, the forced directed placement technique was used to obtain the relative positions of the modules based on the hybrid matrix.

This paper develops a placement scheme which couples both wireability and reliability requirements together for convectively cooled PWBs. Our approach differs from the previous one [Ost90] in two respects. First, we can obtain the Pareto optimum solutions. Second, in our approach, wireability is measured by the maximum wiring density instead of the total wiring lengths. Thus possibility of heavy wiring buildup or congestion regions, especially in the middle of a board, can be avoided.

## 6.1 Problem Description

Consider a two-dimensional board on which the modules are arranged in regular, in-line arrays with the same spacing in both the lateral and longitudinal directions. The modules, representing components or subsystems, are in many typical cases semiconductor chips with external leads to be connected according to some specifications. The board is characterized in terms of a finite array of slots. Each slot is represented by a point in an  $xy$ -coordinate system. Modules are the entities which are to be assigned to the slots to achieve some form of optimum results. Obviously, a module can occupy one and only one slot, and in each slot no more than one module is allowed.

Since a convectively cooled PWB is considered in this study, in order to ease the two-objective optimization process, the above mentioned two-dimensional structure is obtained by folding the corresponding one-dimensional placement result, as shown in Fig. 6.1. That is, the two-dimensional placement problem is treated as a one-dimensional module permutation problem. A one-dimensional module permutation,  $P$ , is expressed as an  $n$ -tuple,  $(e_1, \dots, e_k, \dots, e_n)$ , where the modules,  $e_k$ 's, are ordered by their indices. In the study, the expression of  $(e_{1,1}, e_{1,2}, \dots, e_{r,c}, \dots, e_{R,C})$  is also used, where the subscripts denote the module's position by row and column in the two-dimensional structure, respectively. For example, the module at the second row and second column in Fig. 6.1 can be represented by either  $e_{2,2}$  or  $e_9$ . Note that columns and rows are defined to be parallel and perpendicular to the fluid flow direction, respectively.

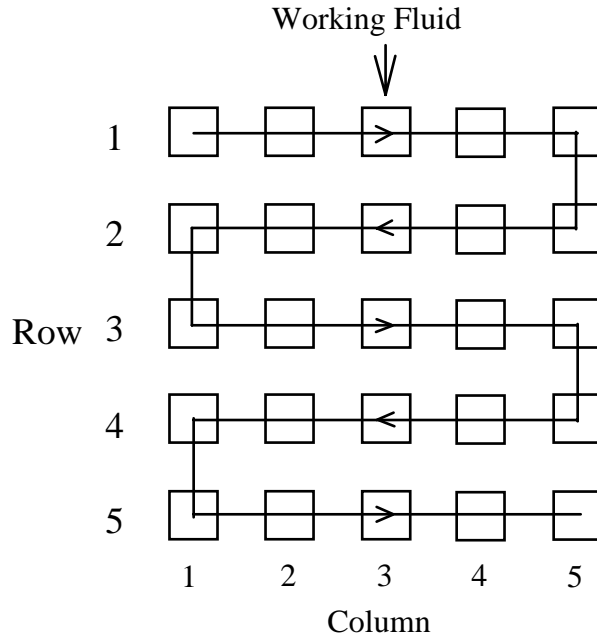


Fig. 6.1 Two-dimensional module placement by folding operation from one-dimensional module placement.

The wireability of any placement  $P$  is measured by the maximum wiring density,  $W(P)$ , which is defined by:

$$W(P) = \mathop{\text{Max}}_{k=1}^{n-1} W(v_k, v_{k+1}),$$

where

$$W(v_k, v_{k+1}) = | \{e_i \mid \exists v_a, v_b \in V(e_i) \ni a \leq k < b.\} | .$$

The reliability is expressed as an Arrhenius-type failure rate model of the form:

$$\lambda(v_k) = B(v_k) \times \exp[-A(v_k) / T_j(v_k)] + E(v_k), \quad (6.1)$$

where  $T_j(v_k)$  and  $\lambda(v_k)$  are the junction temperature and failure rate of  $v_k$ , respectively; the coefficients  $A(v_k)$ ,  $B(v_k)$ , and  $E(v_k)$  are independent of temperature, but are related to device types, quality, other environments, and the production learning curve.

The system failure rate of the board with  $m$  modules can be determined by summing over the failure rates of its constituent modules, if they are connected in series on the board. That is

$$\lambda_T(P) = \sum_{k=1}^m \lambda(e_k). \quad (6.2)$$

The placement problem is now defined as:

*Given a logical diagram, obtain a module permutation  $P$  so as to minimize  $\lambda_T(P)$  and  $D(P)$ .*

No solution satisfies these two objectives simultaneously as stated in Section 1. Only the Pareto optimal solutions can be obtained. Therefore, the problem is restated as:

*Given a logical diagram and the maximum wiring density  $D_{max}$  allowed, obtain a module permutation,  $P$ , so as to minimize  $\lambda_T(P)$ , subject to  $D(P) \leq D_{max}$ . By varying  $D_{max}$  one can obtain all Pareto optimal solutions.*

## 6.2 Junction Temperature Prediction

The junction temperature of  $v_k$ ,  $T_j(v_k)$ , depends on its thermal resistance from junction to case,  $R_{jc}(v_k)$ , the amount of heat dissipated by the module,  $Q(v_k)$ , and the case temperature,  $T_c(v_k)$ . The relationship can be simply represented as:

$$T_j(v_k) = T_c(v_k) + T_{jc}(v_k) = T_c(v_k) + Q(v_k) \times R_{jc}(v_k). \quad (6.3)$$

At steady state  $R_{jc}(v_k)$  is constant, so  $T_j(v_k)$  is primarily a function of  $T_c(v_k)$  and  $Q(v_k)$ .

For steady-state forced convection cooling,  $T_c(v_k)$  depends not only on the heat dissipated by  $e_k$  itself but also on its upstream modules because of thermal wake effects. Since the energy equation is linear, the superposition techniques developed by Arvizu and Moffat, and Anderson and Moffat [Mof90, And92a, And92b] are used to account for the thermal wake effects on downstream modules.

The case temperature of  $v_k$ ,  $T_c(v_k)$ , can be written as the sum of three parts: its temperature rise due to upstream heated modules (i.e. adiabatic temperature rise,  $T_{ad}$ ), its temperature rise due to its own power dissipation (i.e. self heated temperature rise,  $T_{se}$ ), and the inlet temperature of the working fluid,  $T_o$ . That is

$$T_c(v_k) = T_{ad}(v_k) + T_{se}(v_k) + T_o, \quad (6.4)$$

and

$$T_{se}(v_k) = Q(v_k) / [h(v_k) \times A_f], \quad (6.5)$$

where  $A_f$  is the surface area of the module,  $h$  is the forced convection heat transfer coefficient which has to be determined experimentally.

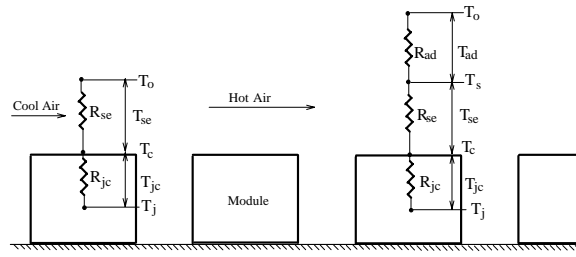


Fig. 6.2 Thermal network for  $T_j$

For  $T_{ad}(e_k)$ , the following two cases are considered.

### 6.2.1 One Module Being Heated

Considering the case of only  $v_{i,j}$  being heated, the ratio of the adiabatic temperature rise of the one-row-downstream module,  $T_{ad}(v_{i+1,j})$ , to the self temperature rise of the heated module,  $T_{se}(v_{i,j})$ , denoted by  $\theta$ , is

$$\theta = T_{ad}(v_{i+1,j}) / T_{se}(v_{i,j}) \quad (6.6)$$

Arvizu and Moffat found that  $T_{ad}$  of the module downstream,  $T_{ad}(v_{r,c})$ , can be expressed by:

$$T_{ad}(v_{r,c}) = \begin{cases} \frac{T_{se}(v_{i,j}) \times \theta}{(r-i)}, & \text{for } c = j; \\ T_{se}(v_{i,j}) \times \theta \times f_r, & \text{for } c = j-1 \text{ or } c = j+1 \\ 0, & \text{otherwise} \end{cases} \quad (6.7)$$

where ' $f_r$ ' is a constant. A typical value of  $f_r$  is about 0.04. The function  $\theta$  directly depends on the maximum velocity turbulence fluctuations [And92], and must be established experimentally, e.g. [Mof92, And92a, And92b, Wiz84, Bib86, Hol87, Cop92].

### 6.2.2 All Modules Being Heated

For the case of all modules being heated,  $T_{ad}(v_{r,c})$  can be determined by the superposition method. That is

$$T_{ad}(v_{r,c}) = \theta \times \sum_{i=1}^{r-1} \left\{ \frac{T_{se}(v_{i,c})}{r-i} + f_r \times [T_{se}(v_{i,c-1}) + T_{se}(v_{i,c+1})] \right\} \quad (6.8)$$

Substituting formulas (6.4), (6.5), and (6.8) into (6.3), one obtains the junction temperature of  $v_{r,c}$  as:

$$T_j(v_{r,c}) = \theta \times \sum_{i=1}^{r-1} \left\{ \frac{T_{se}(v_{i,c})}{r-i} + f_r \times [T_{se}(v_{i,c-1}) + T_{se}(v_{i,c+1})] \right\} + \frac{Q(v_{r,c})}{h(v_{i,c}) \times A_r} + Q(v_{r,c}) \times R_{jc}(v_{r,c}) + T_o. \quad (6.9)$$

The junction temperature for each module is notably independent of the downstream modules in formula (6.9). Therefore, if one assign the modules to slots from upstream to downstream one by one, the junction temperature of each module can be determined once it is assigned to a slot.

### 6.3 Ordered Best-First Search Algorithm

In the study, an ordered best-first search algorithm (OBFSA) is developed to obtain a module permutation with the lowest system failure rate under the constraint of maximum wiring density. This algorithm builds up the linear permutation from front to rear one module at a time based on a priority metric.

#### 6.3.1 Bounding Functions

Three bounding functions are used to test whether the partial permutation being formed has any chance of success.

##### 6.3.1.1 Wiring Density Bound

Let  $P_a$  represent the linear set of the assigned modules,  $\langle v_1, \dots, v_k \rangle$ , and  $P_u$  denotes the sets of unassigned modules. For each assignment process, a module selected from  $P_u$  is assigned to the tail of  $P_a$ . After that  $P_a = \langle v_1, \dots, v_{k+1} \rangle$ . The bounding function of the wiring density for the new partial permutation  $P_a$ ,  $B_{wd}(P_a)$ , is defined by

$$B_{wd}(P_a) = W(v_k, v_{k+1}) - W_{max}.$$

The wiring density bound is true if  $B_{wd}(P_a) \leq 0$ .

##### 6.3.1.2 Violation Number Bound

Let  $P_{best}$  and  $T_{j,av}$  be the best placement currently and the average junction temperature of the modules in  $P_{best}$ , respectively. The bound is based on the following conjecture: if a placement,  $P$ , is better than  $P_{best}$  in terms of the system failure rate, the modules whose junction temperatures are higher than  $T_{j,av}$  in  $P_{best}$  will have lower junction temperatures in  $P$ . For a partial permutation, the number of modules that violate the conjecture is called its violation number. Clearly, a larger violation number implies a small probability to obtain a better complete permutation than  $P_{best}$ .

Formal definition about the violation number of  $P_a$ , denoted by  $VN(P_a)$ , is given by:

$$VN(P_a) = | \{ v_k \mid v_k \in P_a \wedge T_j(v_k^{P_{best}}) > T_{j,av} \wedge T_j(v_k^{P_{best}}) < T_j(v_k^{P_a}) \} |$$

where  $T_j(v_k^{P_{best}})$  and  $T_j(v_k^{P_a})$  are the junction temperatures of  $v_k$  in  $P_{best}$  and in  $P_a$ , respectively.

The bounding function of the violation number,  $B_{vn}(P_a)$ , is defined by

$$B_{v_n}(P_a) = VN(P_a) - VN_{\max},$$

where  $VN_{\max}$  is a critical number. The violation number bound is true if  $B_{v_n}(P_a) \leq 0$ . The best choice of  $VN_{\max}$  is noted to be nearly the square root of the number of modules in the problem as will be shown in Section 6.4.

### 6.3.1.3 Failure Rate bound

The bounding function of the failure rate,  $B_{fr}(P_a)$ , is defined by

$$B_{fr}(P_a) = \lambda_T(P_a) + \lambda_{lb}(P_u) - \lambda_T(P_{\text{best}}),$$

where  $\lambda_{lb}(P_u)$  is the lower bound on the total failure rate of  $P_u$ . The failure rate bound is true if  $B_{fr}(P_a) \leq 0$ . The junction temperature and the failure rate of each module in  $P_u$  are unknown since the module permutation of  $P_u$  is undetermined. Evaluating  $\lambda_{lb}(P_u)$  exactly therefore becomes the key point in using the bound. Two approaches were presented to estimate  $\lambda_{lb}(P_u)$  in the study.

(a) Approach 1:

An obvious lower bound on the junction temperature for any unassigned module,  $v_k$ , can be obtained by ignoring the adiabatic temperature rise and replacing  $h$  with the maximum value,  $h_{\max}$ , in formula (6.9). That is,

$$T_{j,lb}(v_k) = \frac{Q(v_k)}{h_{\max} \times A_r} + Q(v_k) \times R_{jc}(v_k) + T_o \quad (6.10)$$

$T_{j,lb}(v_k)$  is apparently independent of the module placement. The value of  $\lambda_{lb}(P_u)$  therefore can be determined by summing over all of the individual  $\lambda_{lb}(v_k)$  based on  $T_{j,lb}(v_k)$ .

$$\lambda_{lb}(P_u) = \sum_{v_k \in P_u} \lambda_{lb}(v_k) = \sum_{v_k \in P_u} \{E(v_k) + B(v_k) \times \exp[-A(v_k)/T_{j,lb}(v_k)]\}. \quad (6.11)$$

This lower bound is very conservative since the effects caused by the upstream modules are ignored entirely.

(b) Approach 2:

Clearly, the better the estimation of  $\lambda_{lb}(P_u)$  is, the smaller is the solution space searched. The temperature rises caused by one-row-upstream modules are considered in this approach for better estimating of  $\lambda_{lb}(P_u)$  than approach 1. The average value of lower bound on the adiabatic temperature rise of the modules in  $P_u$  can be evaluated as:



$$\overline{T_{ad,lb}}(P_u) = \frac{\theta}{A_r} \sum_{j=0}^c \frac{Q(v_{k-j})}{C \times h(v_{k-j})} + \frac{|P_u| - C}{|P_u|} \sum_{v_k \in P_u} \frac{Q(v_k)}{|P_u| \times h(v_k)} \quad (6.12)$$

where  $v_k$  is the module at the tail of  $P_u$ . The multiplier  $(|P_u| - C) / |P_u|$  is needed since the modules on the final row have no contribution to  $\overline{T_{ad,lb}}(P_u)$ .

Furthermore, by replacing  $h(v_k)$  with the average value,  $\bar{h}$ , in formula (6.12), we obtain

$$\overline{T_{ad,lb}}(P_u) = \frac{\theta}{A_r} \sum_{j=0}^c \frac{Q(v_{k-j})}{C \times h(v_{k-j})} + \frac{|P_u| - C}{|P_u|^2 \times \bar{h}} \sum_{v_k \in P_u} Q(v_k) \quad (6.13)$$

where

$$\bar{h} = \frac{1}{|P_u|} \sum_{v_k \in P_u} h(v_k) \quad (6.14)$$

Formula (6.13) is independent of the placement of the unassigned modules.  $\lambda_{lb}(P_u)$  therefore can be evaluated reasonably by the following expression:

$$\lambda_{lb}(P_u) = \sum_{v_k \in P_u} \{E(v_k) + B(v_k) \times \exp[-A(v_k) / T^j(v_k)]\} \quad (6.15)$$

where

$$T^j(v_k) = T_o + T_{jc}(v_k) + T_{se}(v_k) + \overline{T_{ad,lb}}(P_u) \quad (6.16)$$

Formula (6.13) is apparently improper if only the modules on the final row are unassigned. In such cases, one needs another formula for predicting  $\lambda_{lb}(P_u)$ . Clearly, the adiabatic temperature rise of each unoccupied slot can be determined by formula (6.8) in these cases. An obvious lower bound on the adiabatic temperature rise for unassigned modules therefore is

$$T_{ad,lb} = \min_{v_k \in P_u} T_{ad}(v_k) \quad (6.17)$$

So, the lower bound on the junction temperature of  $v_k$  is

$$T^{''j}(v_k) = T_o + T_{jc}(v_k) + T_{se}(v_k) + T_{ad,lb} \quad (6.18)$$

and the lower bound on the total failure rates of the modules in  $P_u$  is

$$\lambda_{lb}(P_u) = \sum_{v_k \in P_u} \{E(v_k) + B(v_k) \times \exp[-A(v_k) / T^{''j}(v_k)]\}. \quad (6.19)$$

The approach 2 is summarized as follows. During the execution of OBFSA, if more than one row of modules are unassigned,  $\lambda_{lb}(P_u)$  is estimated by formula (6.15); otherwise, it is estimated by formula (6.19).

### 6.3.2 Ordered Best-First Search Algorithm

The initial module permutation for OBFSA is an ordered list which is sorted by the following **priority metric**

$$\frac{d\lambda_{lb}(v_k)}{dT_{j,lb}(v_k)} \times \frac{1}{Q(v_k)}$$

in descendent order, where  $T_{j,lb}(v_k)$  is defined by expression (6.10). The first term in the priority metric is the sensitivity of the failure rate to the junction temperature. The modules with a larger sensitivity to the junction temperature have to be placed closest to the coolant so as to reduce the individual failure rates themselves. On the other hand, those modules with larger heat dissipation rates have to be placed at the rear so that they will not heat other modules. The priority metric combines these two effects together.

OBFSA is initiated by

call OBFSA(1)

The procedure of OBFSA is as follows:

**procedure** OBFSA(k)

global n, P(1:n), P<sub>best</sub>(1:n) // P<sub>best</sub> is the best permutation currently //

**if** k = n **then**

**if**  $\lambda(P) < \lambda(P_{best})$  **then** [P<sub>best</sub>(1:n) ← P(1:n) ; **return**]

**else return**

**endif**

**for** j ← k to n **do**

    MOVE(j,k) // move v<sub>j</sub> into the middle of v<sub>k</sub> and v<sub>k+1</sub> in P(1:n) //

    P<sub>a</sub> ← P(1:k+1)

**if** B<sub>wd</sub>(P<sub>a</sub>) ≤ 0, B<sub>fr</sub>(P<sub>a</sub>) ≤ 0, and B<sub>vn</sub>(P<sub>a</sub>) ≤ 0 **then** call OBFSA(k+1) **endif**

    MOVE(k+1, j) // recover the original permutation in P //

**repeat**

**end** OBFSA

As an example of how OBFSA works, let's consider an initial module permutation, P = {v<sub>1</sub>, v<sub>2</sub>, v<sub>3</sub>, v<sub>4</sub>} which is an ordered list with descendent value of the priority metric. The OBFSA systematically generates the possible permutations by the following procedures:

(1) move v<sub>1</sub> to the first position followed by OBFSA(2) to generate the following permutations of {v<sub>2</sub>, v<sub>3</sub>, v<sub>4</sub>}.

(2) move v<sub>2</sub> to the first position followed by OBFSA(2) to generate the following permutations of {v<sub>1</sub>, v<sub>3</sub>, v<sub>4</sub>}.

(3) move  $v_3$  to the first position followed by OBFS(2) to generate the following permutations of  $\{v_1, v_2, v_4\}$ .

(4) move  $v_4$  to the first position followed by OBFS(2) to generate the following permutations of  $\{v_1, v_2, v_3\}$ .

The expression "followed by OBFS(2)" is the clue to recursion. It implies that one can solve the problem for the set with  $n$  modules if one has an algorithm which works on the  $n-1$  modules.

The list of the unassigned modules is notably still an ordered list with descendent value of the priority metric. The modules with a larger priority value therefore have a prior to be assigned to the front of the placement. This property helps obtain the solution of the lowest system failure rate as fast as possible. Furthermore, it also speeds up the algorithm since the solution space searched is confined by  $\lambda_T(P_{best})$ .

#### 6.4 Computational Results

Three cases are given for the placement procedure presented. The net lists of cases 1 and 3 were from [Sch72] and of case 2 is from [Che87]. For reliability evaluation, the modules are randomly assigned data taken from the MIL-HDBK-217E and are listed in Table 6.1. The data about the adiabatic heat transfer coefficient and the thermal wake functions are from [Mof92]. In all cases the inlet temperature of the working fluid was 25 °C.

In this study, the number of basic operations done (NBOD) is chosen as the complexity measure. Obviously, the basic operation in the OBFS is function MOVE( ).

The Pareto sets for the tested cases are illustrated in Fig. 6.3. They are obtained with approach 1 for estimating  $\lambda_{IB}(P_U)$ . The ranges of system failure rates in Pareto sets were from 2.141 to 2.159 f / M for case 1, 5.775 to 6.034 f / M for case 2, and 16.28 to 25.75 f / M for case 3.

Table 6.1 Some data for the circuits tested.

Items	Case 1: 9 modules, 21 nets, and 57 pins	Case 2: 16 modules, 17 nets, and 42 pins	Case 3: 31 modules, 33 nets, and 79 pins
Placement Type	3 columns by 3 rows	4 columns by 4 rows	5 columns by 7 rows
Package Type	14-lead DIPs	16-lead DIPs	16-lead DIPs
Range of $Q(v_k)$	0.04 to 0.55 W	0.07 to 0.94 W	0.05 to 1.4 W
Sum of $Q(v_k)$	2.58 W	8.33 W	15.87 W
$\theta$	0.114	0.077	0.077
$f_r$	0.04	0.04	0.04

$R_{jc}$ ( $^{\circ}C/W$ )	50	50	50
$h_{ad}$ ( $W/m^2$ $^{\circ}C$ )	55	$135 - 5xr$ *	$135 - 5xr$ for $r < 4$ 115 for $r \geq 4$

\*  $r$  is the row number.

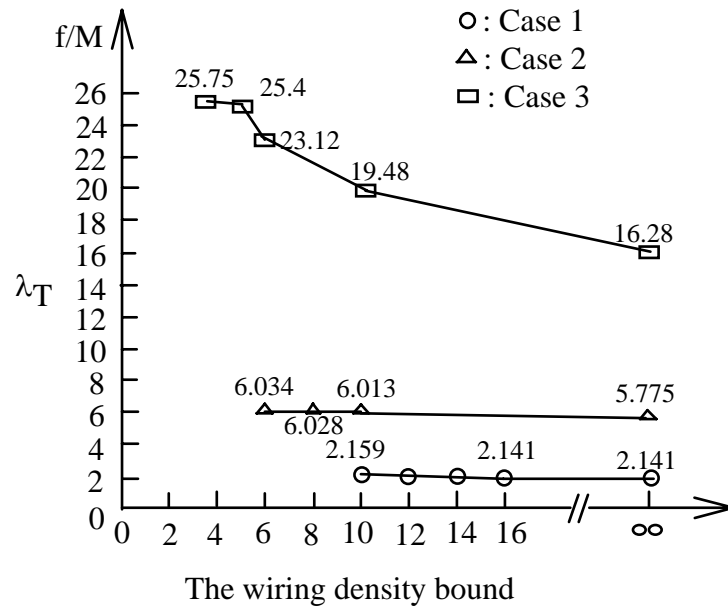


Fig. 6.3 The Pareto curves for the examples tested.

In the following, we focus on the performance of Pareto solutions subject to the minimum wiring density constraint. There are 2212 feasible solutions with the range of the system failure rate from 2.159 to 2.34  $f / M$  for case 1,  $5.66 \times 10^8$  feasible solutions with the range of the system failure rate from 6.034 to 6.553  $f / M$  for case 2, and  $2.87 \times 10^{10}$  feasible solutions with the range of the system failure rate from 25.75 to 30.994  $f / M$  for case 3, as shown in Table 6.2. The above results were obtained by backtracking method with the wiring density bound. The sizes of the solution spaces are observed to be very large. So, the system failure rate can be largely improved without deteriorating the wireability.

Table 6.2 Solution space data with the minimum wiring density bound.

Items	Case 1	Case 2	Case 3
Minimum Wiring Density	10	6	4
No. of Feasible Solutions	2212	$5.66 \times 10^8$	$2.87 \times 10^{10}$
Range of $\lambda_T$ ( $f/M$ )	2.159 to 2.340	6.034 to 6.553	25.750 to 30.994

The first feasible solutions ( $P_1$ ) obtained by OBFSa were compared to the optimum solutions ( $P_{Op}$ ) for all tested cases in Table 6.3.  $P_1$  has a very high quality in system failure rate as expected. The optimization measure of  $P_1$  is more than ninety percentages for all the cases. Furthermore, the NBOD needed for obtaining  $P_1$  is relatively small. For example, the NBOD needed for obtaining  $P_1$  is only 2898 for case 3. This result is very important especially for large-sized problem, since it shows that one can obtain high quality solutions with little computation time by OBFSa.

Table 6.3 Performance of the first feasible solution ( $P_1$ ).  
 $P_{Op}$  denotes the optimum solutions.

	Case 1	Case 2	Case 3
$\lambda_T(P_{Op})$	2.159 f/M	6.034 f/M	25.750 f/M
$\lambda_T(P_1)$	2.171 f/M	6.073 f/M	26.088 f/M
OM( $P_1$ )	93.3 %	92.5 %	93.6 %
NBOD( $P_1$ )	42	554	2898

The reason for a much smaller solution space is due to the application of the violation bound. Its effect is given in Table 6.4. As can be seen, in approach 1, by relaxing the bound, solutions obtained were optimum in system failure rate, if the violation number bound was set to 3 for case 1, 4 for case 2, and 5 for case 3, respectively. The best maximum violation number is noted to be nearly the square root of the problem size. For most cases the results in approach 2 are a little worse than those of approach 1. However, the execution time in approach 2 saves from 2 to 10 times for case 1 and 3, and from 600 to 2400 times for case 2 than in approach 1.

Table 6.4 The effects of the violation number (V.N.).

V.N.	Case 1				Case 2				Case 3			
	Approach 1		Approach 2		Approach 1		Approach 2		Approach 1		Approach 2	
	OM %	NBOD	OM %	NBOD	OM %	NBOD	OM %	NBOD	OM %	NBOD	OM %	NBOD
1	94.5	6968	93.3	1724	92.7	9.2 M	96.0	15.2 K	95.0	52.2M	94.9	28.6 M
2	99.4	10912	93.3	1724	97.3	15.3 M	97.1	10.8 K	96.0	108.8M	95.9	21.2 M
3	100	11052	93.3	1724	98.7	22.9 M	97.1	11.6 K	98.2	144.4M	97.0	18.4 M
4					100	27.7 M	97.1	11.7 K	99.6	133.4M	99.9	28.6 M
5									100	470.0M	99.9	48.6 M

✱ The 'K' and 'M' denote thousand and million, respectively.

One might wonder how effective the present method is over the brute force approach. As an example, for obtaining the Pareto optimum solution under the minimum wiring density bound, the NBOD needed for case 3 is  $1.6 \times 10^{34}$  ( $2n!$ ) for the brute force approach; in contrast, it is only  $4.7 \times 10^8$  and  $4.86 \times 10^7$  for approach 1 and 2 with  $B_{vn}(Pa) \leq 5$ , respectively.

## 6.5 Summary

This paper develops a placement algorithm which couples both reliability and wireability together for arranging electronic modules on a convectively cooled board. The two-dimensional structure is obtained by folding the one-dimensional placement result. The wireability of placement is measured by the maximum wiring density in the one-dimensional placement. The system failure rate can be largely improved without deteriorating the wireability by using the wireability metric because the solution space is very large. An ordered best-first search algorithm is presented for solving the two-objective optimization problem. A priority metric and three bounding functions are applied to guide the search and efficiently prune the solution space. The first solution obtained by the algorithm has very high quality in system failure rates. This is very important especially for large-sized problems, since it infers that one can obtain high quality solutions with less computation time. Moreover, the method is very flexible, if other objectives are needed for placement, they can be included and considered as new bounding functions.