行政院國家科學委員會專題研究計畫　成果報告

子計畫一:半導體需求資料探擷與知識發掘應用於產能配置最佳化之研究(3/3)

計畫類別： 整合型計畫
計畫編號： NSC93-2213-E-002-005-
執行期間： 93 年 08 月 01 日至 94 年 07 月 31 日
執行單位： 國立臺灣大學工業工程學研究所

計畫主持人： 陳正剛

報告類型： 完整報告

處理方式： 本計畫可公開查詢

中　華　民　國 94 年 10 月 5 日

行政院國家科學委員會多年期專題研究計畫期中報告

半導體供應往路決策促成技術研究子計畫三
半導體需求資料探擷與知識發掘應用於產能配置最佳化之研究（3/3）
Semiconductor Demand Data Mining and Knowledge Discovery for Optimization of Capacity Allocation

計畫編號：NSC 90-2218-E-002-046
執行期限：93 年 8 月 1 日至 94 年 7 月 31 日

主持人：陳正剛
(e-mail: achen@ntu.edu.tw)
執行機構及單位名稱：國立台灣大學工業工程學研究所

中文摘要

在需求供給的網路中，需求的不確定性不但會被散播更會在網路中被放大而導致整條供應鏈營運品質低落的連鎖效應。半導體製造網路為最複雜的需求供給網路之一，因此深受不確定需求資料之苦。在上一年度的研究中，我們根據第一年所發掘的知識，探究產能需求群組對於設備整體效能(OEE)的影響，並建構數學模型來描述產能需求配置的影響。第三年的計劃將著重於產能配置的最佳化，研究目標是尋求一以需求群組為基之最佳產能配置，使設備整體效能達到最佳以降低設備產能的需求，並將探討不同的組合最佳化演算法、如貪婪演算法、基因演算法等，最後提出有效的最佳化方法並以半導體的實際需求及製造資料來驗證。

關鍵詞：需求規劃、產能配置、設備整體效能

Abstract

The demand signal is the most unreliable source of information that plagues the operation effectiveness in a demand-supply network. Moreover, the demand uncertainty is not only propagated but also magnified over the network and causes a chain effect on the planning quality of the entire supply chain. Semiconductor manufacturing network is one of the most complicated demand-supply networks and thus suffers greatly from the untrustworthy demand information. In the second year of this project, the effect on the overall equipment effectiveness (OEE) has been explored. The effects of demand grouping for equipment capacity allocation are then modeled mathematically. The model is aimed to help practitioners comprehend how demand plans work together with capacity allocation to affect the OEE. The third year of this project will focus on optimization of capacity allocation. The goal is to find optimum capacity allocation for demand groups to minimize the required equipment capacity or equivalently maximize the OEE subject to uncertain demand signals. Various combinatorial optimization algorithms, such as Greedy Algorithm, Genetic Algorithms, etc., will be investigated. Effective optimization methodologies are then suggested and tested using actual semiconductor demand and manufacturing data.

Keywords：Demand Planning, Capacity Allocation, Overall Equipment Effectiveness (OEE)

1. Introdution

The objective of this year's research is to develop demand grouping strategies for capacity allocation such that the overall equipment effectiveness (OEE) is maximized and thus the capacity required to meet the demand can be minimized.

Overall equipment efficiency is used extensively to quantify the effect of flexibility on equipment efficiency in a manufacturing system. Leachman [2] proposed definitions and mathematical formulas for computing overall efficiency and data collection strategies. The OEE model includes four components [1]:

$$OEE = Availability \times Operating\ Efficiency \times Rate\ Efficiency \times Rate\ of\ Quality$$

The definitions of these components are [3]:
(1) Availability: Up time / Total time
(2) Operating Efficiency: Actual processing time /Theoretical processing time
(3) Rate Efficiency: Run time / Up time
(4)Rate Of Quality: (Total units processed – Total defect units) / Total units processed

The capacity requirement for a type of machine can be then expressed as follows:

$$\text{Capacity Requirement} = \frac{\text{Demand} \times \text{Processing Time}}{\text{Overall Equipment Efficiency}}$$

As can be seen, OEE is a factor inflating required equipment capacity. In this year's research, The goal is to find optimum capacity allocation for demand groups to minimize the required equipment capacity or equivalently maximize the OEE subject to uncertain demand signals.

2. Demand grouping for tool capacity allocation

According to the static capacity models, the capacity demand can be obtained by product demands and processing time:

Capacity demand at time period $t = q_t = d_t \times \tau$

where d_t is the product demand at time period t, t=1,…,T and τ is the processing time required to by one product unit. The capacity requirement is then determined by the average capacity demand and the Overall Equipment Efficiency (OEE):

$$\text{The capacity requirement} = Cr = \frac{\sum_{t=1}^{T} q_t}{T \times o}$$

Where q_t is the capacity demand at time period t, $t=1,...,T$, and o is the overall equipment efficiency

The number of tools must be integer. After capacity requirement is calculated, the tool requirement can be estimated by the capacity requirement. The tool requirement is calculated as follows:

$$Tr = \left\lceil \frac{\sum_{t=1}^{T} q_t}{T \cdot o \cdot Cpm} \right\rceil$$

where Cpm is the capacity provide by one tool at one time period

The objective is to find the best way of grouping among these five possible options. To answer this question, we first develop a matrix form to express these different grouping types. We build a matrix with columns representing the products and rows representing the groups. Since in this example, there are 3 products, that can be grouped into 3 groups at most, we build a 3×3 matrix as follows:

$$group \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \overset{product}{\overset{(1 \quad 2 \quad 3)}{\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}}}$$

$$\text{where } x_{ij} = \begin{cases} 1, & \text{if product i is grouped into group } j \\ 0, & \text{otherwise} \end{cases}$$

To ensure the 1-to-1 mapping between the assignment matrix and the actual assignment, we let the group number to be and only be the smallest product number in the group. In other words, when product 1, 2 & 3 are all grouped together, the smallest product number is 1, and so is the group number. The steps to encode grouping type into a machine group matrix are:

1. Assign numbers to products
2. Choose the smallest product number in each machine group as the machine group number.
3. Build an *n×n machin- group matrix* **M**.

$$\mathbf{M} = machine\ group \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix} \overset{product}{\overset{(1 \quad 2 \quad \cdots \quad n)}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{pmatrix}}}$$

$$\text{where } x_{ij} = \begin{cases} 1, & \text{if product i allocated to machine group j} \\ 0, & \text{otherwise} \end{cases}$$

subject to the following constraints:

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad (1)$$

$$x_{ii} = 1 \text{ and } x_{ij} = 0 \text{ for j} < \text{i} \quad \text{when } \sum_{j=1}^{n} x_{ij} \neq 0 \qquad (2)$$

Constraint (1) is ensure that each product can be only assigned to one machine group. Constraint (2) avoids illegal matrices. $\sum_{j=1}^{n} x_{ij} \neq 0$ means there are products in machine group i so that i should be the smallest product number in this machine group; i.e. $x_{ii} = 1$ and $x_{ij} = 0$ for $j < i$. We then define a *capacity-demand-group matrix* **D**:

$$\mathbf{D} = group \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix} \overset{product}{\overset{(1 \quad 2 \quad \cdots \quad n)}{\begin{pmatrix} D_{11} & D_{12} & \cdots & D_{1n} \\ D_{21} & D_{22} & \cdots & D_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ D_{n1} & D_{n2} & \cdots & D_{nn} \end{pmatrix}}}$$

$$\text{where } D_{ij} = \begin{cases} C_i, & if\ x_{ij} \text{ in matrix } \mathbf{M} \text{ is } 1 \\ 0, & if\ x_{ij} \text{ in matrix } \mathbf{M} \text{ is } 0 \end{cases}$$

and C_i is capacity demand of product i, $i = 1,...,n$

Recall that **M** is the machine-group matrix defined earlier. If machine group matrix is

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$

then capacity demand group matrix will be

$$\mathbf{D} = \begin{pmatrix} C_1 & 0 & 0 \\ 0 & C_2 & C_3 \\ 0 & 0 & 0 \end{pmatrix}.$$

3. Minimizing Tool Requirement by Maximizing OEE

The overall equipment efficiency (OEE) measures four components of equipment performance:

Overall Equipment Efficiency

$$= Availability \times RateEfficiency \times RateOfQuality \times OperatingEfficiency$$

$$= \frac{\text{Up Time}}{\text{Total Time}} \times \frac{\text{Product Time}}{\text{Up Time}} \times \text{Yield} \times \frac{\text{Theoretical Processing Time}}{\text{Actual Processing Time}}$$

$$= Availability \times Utilization \times Yield \times Efficiency$$

The OEE will be influenced by machine allocations to product groups. In this section we try to model the impacts of product grouping for machine allocation on the OEE.

The first objective of our research is to minimize the average capacity requirement to meet the demand. The capacity requirement can be calculated as follows:

Average capacity requirement for machine group i

$$= Cr_i = \frac{\text{Average capacity demand}}{\text{Overall Equipment Efficiency}} = \frac{\overline{q}_i}{o_i},$$

where $\overline{q}_i$ is the average capacity demand of machine group i, and o_i is the overall equipment efficiency of machine group i

And the average tool requirement for machine group i can be calculated by the following formula:

$$Tr_i = \left\lceil \frac{Cr_i}{Cpm} \right\rceil$$

where Cpm is the capacity of one tool per time period and q_{it} is the capacity demand of machine group i in time period t. Then the quantity of average tool requirement is

$$ATR = \sum_i Tr_i \qquad (3)$$

Since the capacity demand will fluctuate according to product demand fluctuation, the tool requirement will surely also fluctuate as well. When a fixed quantity of tools is prepared, the number of tools may be insufficient or oversupplied due to the demand fluctuation. Here, we use the sum of squared errors (*SSE*) to measure the tool requirement variability. By product grouping, our second attempt is to minimize the tool undersupply and oversupply errors.

Actual tool requirement of group i in each time period can be calculated as:

$$\frac{\bar{q}_i / o_i}{Cpm},$$

where Cpm is the capacity of one tool per time period. And the quantity of tool requirement of machine group i.

$$Tr_i = \left\lceil \frac{Cr_i}{Cpm} \right\rceil$$

So the squared errors of tool preparation in machine group i will be

$$\mathrm{SE_i} = \sum_{t=1}^{T} \left(\frac{\mathrm{q_{it}} / o_i}{Cpm} - Tr_i \right)^2$$

And we set the sum of squared errors of tool requirement as the second objective:

$$SSE = \sum_i \mathrm{SE_i} = \sum_i \sum_{t=1}^{T} \left(\frac{\mathrm{q_{it}} / o_i}{Cpm} - Tr_i \right)^2 \qquad (4)$$

3.1 Greedy Algorithms for Tool Capacity Allocation

The idea of the Greedy algorithm for minimizing average tool requirement (*ATR*) is presented in the following steps:

1. Compute the *ATR* of all product demand separated.
2. Compute the *ATR* of all possible grouping types in which only a pair of product demands is grouped.
3. If a lower *ATR* can be found many of the grouping types in Step 2, choose the grouping type with the lowest *ATR* and aggregate the pair of product demands in this chosen grouping type. Treat the aggregated product demand as a single product demand. If no *ATR* reduction can be found, stop the process and a solution is reached.
4. Go to Step 2

The idea of the Greedy algorithm for the sum of squared errors (*SSE*) is presented in the following steps:

1. Compute the *SSE* of all product demand separated.
2. Compute the *SSE* of all possible grouping types in which only a pair of product demands is grouped
3. If a lower *SSE* can be found many of the grouping types in Step 2, choose the grouping type with the lowest *SSE* and aggregate the pair of product demands in this chosen grouping type. Treat the aggregated product demand as a single product demand. If no *SSE* reduction can be found, stop the process and a solution is reached.
4. Go to Step 2

Figure 1 shows the flow chart will illustrating the above procedure, the objective value is *ATR* or *SSE* depends on which is used as objective functions.

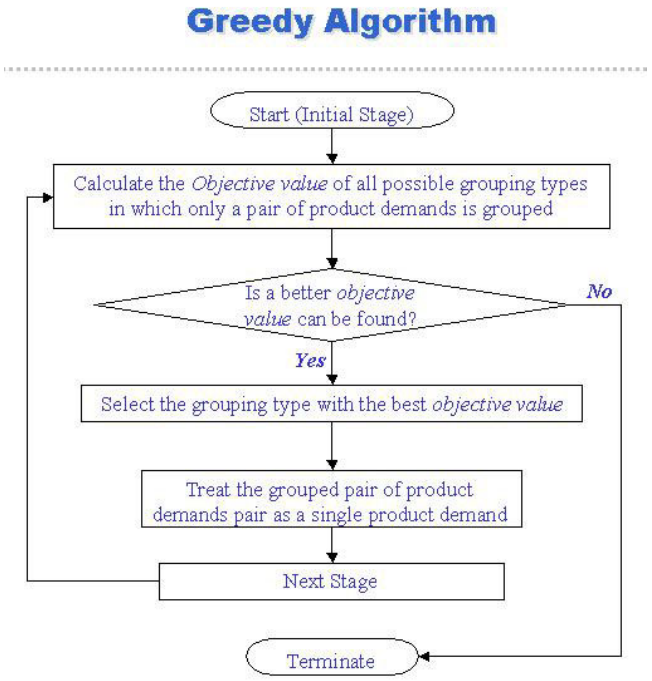

Figure 1 Flow of the Greedy algorithm

3.2 Genetic Algorithms for Tool Capacity Allocation

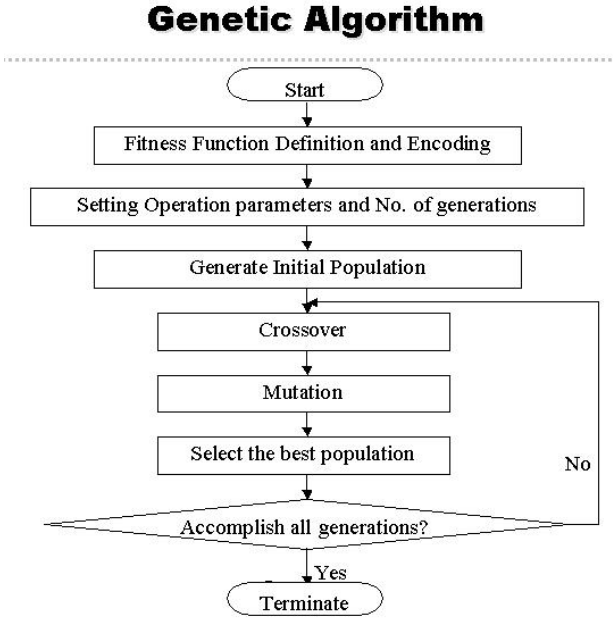

Figure 2 Flow of the Genetic algorithm

This section will illustrate how the Genetic algorithm (GA) will be applied. We will always throw away the worst

chromosomes ("**only the fittest survive**") to encourage the best solutions generated and keep the population **improving** as a whole. In the Greedy algorithm, we must compute many possible solutions in each stage and then select the best one, Genetic algorithm is anticipated to be more efficient to obtain a relatively good solution. But it also can't guarantee to find the optimum solution. The procedure of GA is shown in Figure 2. The details of each step are now described below.

The two objective functions (3) and (4) will be taken as the two fitness functions in the Genetic algorithm: one is the **average tool requirement (*ATR*)** and the other is **Sum of Squared Errors of Tool Requirement (*SSE*)**. In order to solve the problem by the Genetic Algorithm, the solutions of the problem have to be encoded as chromosomes first. Now, the **machine group matrix** developed in the second year of this project and also briefly described earlier can be regarded as one chromosome because it represents one of the solutions (i.e. grouping type) to the product grouping problem.

The crossover operator is an n-point operator that randomly selects two chromosomes from the population called parents, and then joins together the "non-corresponding" parts of each parent to obtain the offspring. The simplest crossover is one-point crossover; the following example will explain how the one-point crossover operates. Chromosomes A and B are randomly selected from the population as parents, and column j is randomly chosen to be the breakpoint; i.e. the parent chromosomes will be broken between column j and j+1 and form the front and back segment:

chromosome A=

$$\left[\begin{array}{ccc:ccc} a_{1,1} & \cdots & a_{1,j} & a_{1,j+1} & \cdots & a_{1,n} \\ a_{2,1} & \cdots & a_{2,j} & a_{2,j+1} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & \cdots & a_{n,j} & a_{n,j+1} & \cdots & a_{n,n} \end{array}\right]$$

chromosome B=

$$\left[\begin{array}{ccc:ccc} b_{1,1} & \cdots & b_{1,j} & b_{1,j+1} & \cdots & b_{1,n} \\ b_{2,1} & \cdots & b_{2,j} & b_{2,j+1} & \cdots & b_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{n,1} & \cdots & b_{n,j} & b_{n,j+1} & \cdots & b_{n,n} \end{array}\right]$$

By crossing over, two offspring will be obtained. We combine the front segment of chromosome A and the back segment of chromosome B to be the front and back segments of one child chromosome, and combine the front segment of chromosome B and back segment of chromosome A to be the front and back segments of the other child chromosome. So the two children chromosomes are as following:

chromosome 1=

$$\left[\begin{array}{ccc:ccc} a_{1,1} & \cdots & a_{1,j} & b_{1,j+1} & \cdots & b_{1,n} \\ a_{2,1} & \cdots & a_{2,j} & b_{2,j+1} & \cdots & b_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & \cdots & a_{n,j} & b_{n,j+1} & \cdots & b_{n,n} \end{array}\right]$$

chromosome 2=

$$\left[\begin{array}{ccc:ccc} b_{1,1} & \cdots & b_{1,j} & a_{1,j+1} & \cdots & a_{1,n} \\ b_{2,1} & \cdots & b_{2,j} & a_{2,j+1} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{n,1} & \cdots & b_{n,j} & a_{n,j+1} & \cdots & a_{n,n} \end{array}\right]$$

After crossing over, chromosomes are broken between columns, all the columns will still contains one entry equal to 1 and others equal to zero. That means every product is still allocated to only one machine group. Both the children inherit segments of both parents' genetic information, so that children may possibly combine the good segments from the parents to obtain a better quality chromosome.

Since there is no reason why the center of the chromosome should be preferred over its ends, a two-point crossover is even more appealing. And in many literatures, two-point crossover we usually consider performs better than others. Also, from the results of our tests, the two-points crossover indeed performs better than the one-point and three-point crossover.

On the basis of above descriptions, the two-points crossover will be applied in this study. The operation of the two-point crossover is similar to that of one-point crossover. Chromosomes A and B are randomly selected from the population as parents and column i and j are randomly chosen to be the breakpoints and split the chromosomes into three segment:

chromosome A =

$$\left[\begin{array}{ccc:ccc:ccc} a_{1,1} & \cdots & a_{1,i} & a_{1,i+1} & \cdots & a_{1,j} & a_{1,j+1} & \cdots & a_{1,n} \\ a_{2,1} & \cdots & a_{2,i} & a_{2,i+1} & \cdots & a_{2,j} & a_{2,j+1} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & \cdots & a_{n,i} & a_{1,i+1} & \cdots & a_{n,j} & a_{n,j+1} & \cdots & a_{n,n} \end{array}\right]$$

chromosome B =

$$\left[\begin{array}{ccc:ccc:ccc} b_{1,1} & \cdots & b_{1,i} & b_{1,i+1} & \cdots & b_{1,j} & b_{1,j+1} & \cdots & b_{1,n} \\ b_{2,1} & \cdots & b_{2,i} & b_{2,i+1} & \cdots & b_{2,j} & b_{2,j+1} & \cdots & b_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{n,1} & \cdots & b_{n,i} & b_{1,i+1} & \cdots & b_{n,j} & b_{n,j+1} & \cdots & b_{n,n} \end{array}\right]$$

Since the parent chromosomes joins together the "non-corresponding" parts of each parent to obtain the offspring in the crossover operation, one of the children is formed by the first segment of chromosome A, the second segment of chromosome B, and the third segment of

chromosome A; the other child chromosome is formed by the first segment of chromosome B, the second segment of chromosome A, and the third segment of chromosome B:

chromosome 1 =

$$\left[\begin{array}{ccc:ccc:ccc} a_{1,1} & \cdots & a_{1,i} & b_{1,i+1} & \cdots & b_{1,j} & a_{1,j+1} & \cdots & a_{1,n} \\ a_{2,1} & \cdots & a_{2,i} & b_{2,i+1} & \cdots & b_{2,j} & a_{2,j+1} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & \cdots & a_{n,i} & b_{1,i+1} & \cdots & b_{n,j} & a_{n,j+1} & \cdots & a_{n,n} \end{array}\right]$$

chromosome 2 =

$$\left[\begin{array}{ccc:ccc:ccc} b_{1,1} & \cdots & b_{1,i} & a_{1,i+1} & \cdots & a_{1,j} & b_{1,j+1} & \cdots & b_{1,n} \\ b_{2,1} & \cdots & b_{2,i} & a_{2,i+1} & \cdots & a_{2,j} & b_{2,j+1} & \cdots & b_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{n,1} & \cdots & b_{n,i} & a_{1,i+1} & \cdots & a_{n,j} & b_{n,j+1} & \cdots & b_{n,n} \end{array}\right]$$

For the mutation operation of GA, we will randomly choose chromosomes from the population to perform mutation operation. The mutation operation corresponds to randomly select some products and allocate them to other randomly chosen machine groups. For example, the following chromosome is chosen to operate mutation; product *j* is grouped into machine group 1 originally (i.e. $a_{j,1} = 1$; $a_{j,2}, \cdots, a_{j,n} = 0$).

$$\begin{bmatrix} a_{1,1} & \cdots & 1 & \cdots & a_{1,n} \\ a_{2,1} & \cdots & 0 & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & \cdots & 0 & \cdots & a_{n,n} \end{bmatrix}$$

If the mutation operation shifts the product j form machine group 1 to machine group n, then the chromosome becomes:

$$\begin{bmatrix} a_{1,1} & \cdots & 0 & \cdots & a_{1,n} \\ a_{2,1} & \cdots & 0 & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & \cdots & 1 & \cdots & a_{n,n} \end{bmatrix}$$

One goal of encoding grouping problem into products group matrix is to avoid illegal machine group matrices (chromosomes). But after crossover and mutation operation, the chromosomes might become illegally. For example, operating crossover:

chromosomes A= $\left[\begin{array}{cc:c} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{array}\right]$ and B = $\left[\begin{array}{cc:c} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{array}\right]$

results in:

chromosome 1 = $\left[\begin{array}{cc:c} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array}\right]$ and 2 = $\left[\begin{array}{cc:c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right]$

But chromosome 1 is an illegal machine group matrix since the machine group 2's smallest product number is not 2 but 3. A legal matrix should be:

chromosome 3 = $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

So, chromosome 1 must be modified to chromosome 3. The steps of machine group modification are:

1. Current row is the first row.
2. If the current row is a zero row or the first nonzero entry in the current row is x_{ii}, go to step 4. If the first nonzero entry in the current row is x_{ij} and $i \neq j$, go to step 3.
3. Exchange row *i* and row *j*, go to step 2.
4. If the current row is the last row, then STOP. Otherwise check the next row; that is let the next row be the current row, and go to step 2.

According to the two-point crossover rules, products in the same group might be separated after crossover. Two neighboring products with numbers in the center of the product number serials will have more chance to be separated after crossover, and two neighboring products with numbers in the two ends of the product number serials will more likely to remain in the same machine group after crossover.

From the results of our 1st-year research and basic statistical inference, it is known that the more complementary the demand correlation of two demand sources ($\rho \rightarrow -1$), the lower the variance of the aggregated demand. The idea of the assignment method is to increase the opportunity for two products with positive correlation to be separated and to avoid two products with negative correlation from being separated. So two products with more complementary correlation ($\rho \rightarrow -1$) will be assigned numbers in the two ends of the product number serials, and two products with less complementary correlation ($\rho \rightarrow 1$) will be assigned numbers in the center of the product number serials.

If there are n products, computing correlations of all product pairs first. Then, assign the two products with the most complementary correlation number 1 and number 2; assign those with the second complementary correlation number n and number n-1; those with the third complementary correlation number 3 and number 4; those with the forth smallest correlation number n-2 and number n-3, and so on. Figure 3 illustrates the efficient product number assignment.

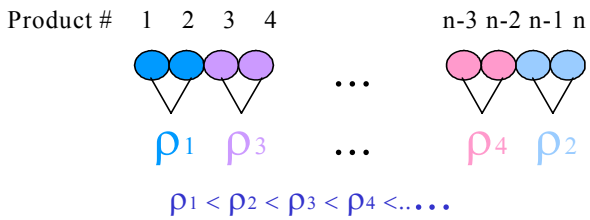

Figure 3 Product number assignment for efficient

4. Case Study: Semiconductor Demand Data

A set of real demand data is used for verification and evaluation of the two proposed algorithms, GA algorithm and Greedy algorithm. There are 300 product demand records and 65 critical machines in the data. The data also recorded the processing time of the machines each product has to be processed by. One of the 65 machines is analyzed. There are 60 products having to process on this machine, and we obtained the capacity demand of these 60 products by their demand and processing time on this machine. The backgrounds of the 60 products' capacity demand data are listed as follows:

1. Time horizon: 69 weeks
2. Capacity demand: in minutes.
3. Products are represented in numbers; each unique number represents a specific product.

The *Bell number* of 60 products is $B(n)=9.769\times10^{59}$, that means the total number of possible combinations for 60 products is equal to 9.769×10^{59}, which is enormous for finding an optimal combination..

Before applying these algorithms, the products should be assigned serial numbers, the products number assignment method was presented in the Section 3.5.6, the idea of this method is assigning two products with more complementary correlation ($\rho\rightarrow-1$) numbers in the two ends of the product number serials, and assigning two products with less complementary correlation ($\rho\rightarrow1$) numbers in the center of the number serials. Since there are 60 different products, we will assign number 1~60 to these products. For example, product 265 and product 149 are the most complementary pair among the 60 products with correlation coefficient of their demands equal to –0.35613. Thus, number 1 and number 2 are assigned to the two products. Similarly, product 145 and product 213 are the second most complementary pair with the demand correlation equal to –0.33030 and, therefore, the two products are assigned number 59 and number 60, respectively.

In addition to the two objective function, *ATR* and *SEE*, three other performance measures, *STR*, *SV* and *CumpuTime,* will be used to evaluate the results of proposed algorithms:

1. *Average tool requirement (ATR)*
2. *Sum of squared errors of tool requirement (SSE)*
3. *Standard error of tool requirement (STR)*

$$STR=\sqrt{\frac{\sum_i SE_i}{T-1}}=\sqrt{\frac{\sum_i\sum_{t=1}^{T}\left(\frac{q_{it}/o_i}{Cpm}-Tr_i\right)^2}{T-1}}$$

4. *CV of tool requirement (CV)*

$$CV=\frac{STR}{ATR}=\frac{\sqrt{\frac{\sum_i\sum_{t=1}^{T}\left(\frac{q_{it}/o_i}{Cpm}-Tr_i\right)^2}{T-1}}}{\sum_i\left[\frac{Cr_i}{Cpm}\right]}$$

5. Computation Time: *CompuTime*

In the literature, two-points crossover is usually considered better than others. After some tests, we find that the population size doesn't have significant effect on the performance of GA. In this research, we set a population size to 4 and the number of breakpoints to 2. This setting performs better, though not significantly, than other setting in our empirical tests. The number of generation is the terminate function in the algorithm; the searching process will stop after the pre-determined number of generations is reached. The probability of crossover is set to 0.9 and the probability of mutation is set to 0.2.

Since we have no OEE data to analyze how the **yield group matrix** and **efficiency group matrix** should be, in this research the **yield group matrix** and **efficiency group matrix** are randomly generated. And the values of OEE model parameters are chosen empirically. Two models of OEE parameters settings will be tested to investigate the impacts of OEE on grouping results. In the *OEE model 1*, the equipment is assumed to be relatively inefficient with OEE around 0.45~0.55 while in the *OEE model 2,* the OEE is assumed to be high (0.6~0.7).

OEE Model Parameters settings:

※ *OEE model 1:*

1.Utilization Model:
- *(1) Utilization upper bound (U): 0.9*
- *(2) Utilization lower bound (L): 0.8*
- *(3) Utilization enhancing factor (r): 0.36788*

2.Yield Model
- *(1) Yield decreasing factor (q): 0.95*
- *(2) Yield heterogeneity penalty factor (p): 0.98*

3.Efficiency Model: *Changeover time (t): 500*

※ *OEE model 2:*

1.Utilization Model:
- *(1) Utilization upper bound (U): 0.99*
- *(2) Utilization lower bound (L): 0.92*
- *(3) Utilization enhancing factor (r): 0.36788*

2.Yield Model
- *(1) Yield decreasing factor (q): 0.98*
- *(2) Yield heterogeneity penalty factor (p): 0.999*

3.Efficiency Model: *Changeover time (t): 200*

The range of OEE in *OEE model 1* is about 0.45~0.55. The range of OEE in *OEE model 2* is about 0.6~0.7.

To evaluate the performance of the two proposed algorithms, we use two types of grouping as the benchmarks: one is grouping all products together, i.e., all products are allocated to the same machine groups, and the other is one-product-one-group, i.e., different products are allocated

to different machine groups.

Results under the two OEE models will be first summarized in Tables 1 and 2. The performance of the algorithms under different OEE models then will be discussed.

Table 1 Comparison of the performance of the algorithms in *OEE model 1*

Algorithms	All Grouped	All Separated	ATR-Greedy	SSE-Greedy	ATR-GA	SSE-GA
SSE	19626626	7135013	5980323	5937320	6384412	5992714
STR	541	326	299	298	309	299
ATR	670	533	444	448	462	467
CV	0.808	0.612	0.673	0.664	0.668	0.640
CompuTime	1 second	3 seconds	2177 Mins	3056 Mins	97 Mins	138 Mins

Table 1 Comparison of the performance of the algorithms in *OEE model 2*

Algorithms	All Grouped	All Separated	ATR-Greedy	SSE-Greedy	ATR-GA	SSE-GA
SSE	7481124	5395973	604172	4695190	5331871	4708410
STR	334	284	300	265	282	265
ATR	414	471	395	396	409	417
CV	0.807	0.603	0.760	0.668	0.690	0.636
CompuTime	2 second	3 seconds	2086 Mins	3183 Mins	113 Mins	175 Mins

In summary, when OEE is higher, it should take *SSE* as an objective function; when OEE is lower, it should take *ATR* as the objective function. For the short-term planning that require responsive solutions, Genetic algorithms will be applied more efficiency; for the long-term planning that emphasize more on the quality than on the speed, Greedy algorithms can provide better solutions.

5. Conclusions

Grouping for machine capacity allocation is a complex combinatorial problem. This research formulated the problem and developed the Greedy and Genetic algorithms to provide the strategies of product demands grouping. Appropriate demand grouping can significantly reduce the average quantity and variability of the tool requirement by improving the OEE and reducing the demand uncertainty. The conclusions of this research are summarized as follows.

We used two objective functions in this research. The first is the quantity of average tool requirement (*ATR*), the second is the sum of squared errors (*SSE*) of tool requirement to measure the quantity and the variability of the tool requirement respectively.

1. When the OEE is low, both objective functions minimize the average tool requirement and the sum of squared errors similarly. The Greedy algorithms provide better results than the Genetic algorithms by both the *ATR* and the *SSE*.
2. When the OEE is high, both objective functions minimize the average tool requirement, but only the objective functions, *SSE*, can significantly reduce the sum of squared errors of tool requirement. The Greedy algorithms also provide better results than the Genetic algorithms by both the *ATR* and the *SSE*.

Comparing the computing efficiency of the two algorithms, though the Greedy algorithms provide better grouping results, it takes about 20 times as much computing time as the GA does. The Genetic algorithms provide good grouping results in a much shorter time.

References

[1] Witte, J. D., "Using Static Capacity Modeling Techniques in Semiconductor Manufacturing", IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop, pp.31-35, 1996.

[2] Leachman, R. C., "Closed-Loop Measurement of Equipment Efficiency and Equipment Capacity", IEEE Semiconductor Manufacturing Technology Workshop, pp.84-90, 2000.

[3] Sattler, L., "Semiconductor Metrics: Conflicting Goals or Increasing Opportunities", IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp.115-126,1995.

[4] Kao, Chung-En, and Chou, Y-C, "A Tool Portfolio Planning Methodology for Semiconductor Wafer Fabs", IEEE Semiconductor Manufacturing Technology Workshop, pp. 84-90, 2000.