International Journal of Production Research

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

Route planning for two wafer fabs with capacity-sharing mechanisms

Muh-Cherng Wu [a] , Chen-Fu Chen [a] & Chang-Fu Shih [a]

[a] Department of Industrial Engineering and Management , National Chiao Tung University , Hsin-Chu, Taiwan, ROC

Published online: 24 Jul 2009.

To cite this article: Muh-Cherng Wu , Chen-Fu Chen & Chang-Fu Shih (2009) Route planning for two wafer fabs with capacity-sharing mechanisms, International Journal of Production Research, 47:20, 5843-5856, DOI: 10.1080/00207540802172029

To link to this article: http://dx.doi.org/10.1080/00207540802172029

International Journal of Production Research
Vol. 47, No. 20, 15 October 2009, 5843–5856

Taylor & Francis
Taylor & Francis Group

Route planning for two wafer fabs with capacity-sharing mechanisms

Muh-Cherng Wu*, Chen-Fu Chen and Chang-Fu Shih

Department of Industrial Engineering and Management, National Chiao Tung University, Hsin-Chu, Taiwan, ROC

(*Received 20 July 2007; final version received 18 December 2007*)

This paper formulates and solves a route planning problem for semiconductor manufacturing. In order to quickly respond to rising demand, a semiconductor company usually adopts a *dual-fab* strategy to expand capacity. That is, two fab sites are built as neighbours and can easily share capacity. Through the capacity-sharing design, a product may be produced by a *cross-fab route*. That is, some operations of a product are carried out in one fab and the other operations in the other fab. This leads to a routing planning problem, which involves two decisions – determining the cut-off point of the cross-fab route and the route ratio for each product – in order to maximise the throughput subject to a cycle time constraint. An *LP-GA* method is proposed to solve the route planning problem. We first use the LP module to make the cut-off point decisions, and proceed to use the GA module for making the decision on the route ratio. Experimental results show that the *LP-GA* method significantly outperforms other methods.

Keywords: operations management; operations strategy; operations planning

1. Introduction

The semiconductor manufacturing industry has to consider the following three factors when expanding capacity. The equipment costs are very expensive, perhaps costing over one billion dollars for a 12 inch wafer fab. The lead time for equipment acquisition is quite long, ranging from 3 to 9 months. However, building the factory space is relatively inexpensive, but with a much longer lead time – taking about one to two years.

In order to respond quickly to rising demand, a semiconductor company usually adopts a *dual-fab* strategy to expand capacity. That is, a large-scale factory space that can accommodate two fabs is established in advance. Then, the equipment for the two fabs is gradually moved into the space over time according to market demand. The two fabs, close to each other in location, support each other and should be managed in an integrated manner.

In such a dual-fab configuration, a relatively easy way to manage is to manufacture each wafer job in one fab. That is, each fab is run separately, without any mutual support in capacity. Such a *separated-operation* paradigm usually leads to an underutilisation of equipment. To remedy the underutilisation issue, a *cross-fab production* paradigm is proposed. This means that a wafer job is partly manufactured in one fab and partly manufactured in the other fab.

*Corresponding author. Email: mcwu@cc.nctu.edu.tw

ISSN 0020–7543 print/ISSN 1366–588X online

DOI: 10.1080/00207540802172029
http://www.informaworld.com

Such a cross-fab production paradigm yields a route planning problem – how to appropriately assign the operations of a wafer job to each of the two fabs. Only a few studies on the route planning problem have been published. Toba *et al.* (2005) addressed the route planning problem in a real-time manner. That is, whenever an operation of a job is completed, a decision – which fab to use to manufacture the next operation – must be made immediately. Wu and Chang (2007) investigated the route planning problem in a short-term or weekly manner, in which the two fabs exchange capacity weekly to maximise the total throughput.

Although having established significant milestones, these two studies have some limitations due to their implicit assumption. They both assume that the transportation times within a fab or among fabs are constant. This implies that the transportation capacity is infinite, and the route planning algorithm may yield a solution with too much transportation. This may lead to traffic jams and, as a result, may lower the throughput and lengthen the cycle time.

In semiconductor manufacturing, the wafer size has steadily increased over time. In a modern fab (12 inch wafer fab), wafer jobs must be transported by automatic vehicles because a wafer job weighs about 30 kg and cannot be handled manually. This may yield a traffic jam problem because the transportation capacity is limited. Our interview with practitioners indicates that the traffic jam symptom occurs particularly with a dual-fab layout. Therefore, the transportation capacity has to be considered in the route planning problem for a modern fab.

This research investigates the route planning problem for a dual-fab layout and is unique in two ways. First, we assume that the transportation capacity is finite and the transportation times vary. Second, the route planning decision is made based on a relatively longer time horizon, for example one or several months. This research, focusing on a relatively long-term decision, complements prior studies which focused on either short-term or mid-term decisions on route planning.

The remainder of the paper is organized as follows. Section 2 reviews the literature relevant to this research. Section 3 presents the route planning problem in detail. Section 4 describes the solution framework, including a linear programming (LP) model, a binary search algorithm, a queuing network model, and a genetic algorithm (GA). Section 5 describes the LP model and the binary search algorithm. Section 6 describes the queueing model and the GA. Numerical experiments are presented in Section 7 and the concluding remarks are in the final section.

2. Relevant literature

Given customer demand, more than one manufacturing site may exist to fulfill that demand. The decision problem is how to allocate the demand to each manufacturing site. This capacity allocation problem can be addressed either at the product level or the operation level.

For the problem at the product level, each site is designated to manufacture a set of products. This implies that a product should be completely manufactured within a single site – cross-site production is prohibited. At the operation level, each site is designated to carry out a group of operations. Then, the operations for manufacturing a product can be distributed among different sites – cross-site production is allowed. This leads to the need to study the *route-planning problem*.

For the capacity allocation problem – *without any cross-site routes*, Wu *et al*. (2005) have given a comprehensive survey. Some recent studies are listed (Rupp and Ristic 2000, Frederix 2001, Karabuk and Wu 2003, Manmohan 2005, Lee *et al*. 2006, Chiang *et al*. 2007). Linear programming models are commonly used to solve the problems. To address the interactions among manufacturing sites, game theory was proposed to enhance the LP model (Mieghem 1999).

For the capacity allocation problem – *with some cross-site routes*, most studies have addressed the problem in the context of group technology (GT). That is, each site is a manufacturing cell and multiple cells form a factory. Cross-cell production for manufacturing a product is permitted. However, each product is preferably manufactured within a particular cell and cross-cell production should be minimised.

Most prior studies allocated the capacity demand to cells by solving a cell formation problem (Avonts and Wassenhove 1988, Kim *et al*. 2005, Vin *et al*. 2005, Dimopoulos 2006, Mahdavi *et al*. 2006, Nsakanda *et al*. 2006, Spiliopoulos and Sofianopoulou 2007). That is, in order to minimise the number of cross-cell transportations, researchers have to answer the question: how many cells should be formed and how should each cell be equipped? After the cell formation problem has been solved, each product is assigned to a particular cell for handling most of its operations. The remaining operations, much fewer in number, are handled by other cells. A GT cell is designed for manufacturing a particular group of products, and by nature is limited in its functional capacity. Therefore, cross-cell routes are unavoidably demanded in GT in order to enhance its functional spectrum.

However, in the route-planning problem addressed here, each of the two fabs is assumed to be functionally comprehensive. That is, a product can be completely manufactured in either one of the two fabs. The purpose of cross-fab production is to increase the total throughput of the two fabs, with the rationale explained below.

In practice, a semiconductor fab is equipped to fulfill the demand of a particular product mix, which is generally obtained from the demand forecast at the time of purchasing the equipment. However, the market demand in terms of product mix may change over time. Therefore, a fab may be underutilised due to a change of product mix. In addition, the two fabs, even if both functionally comprehensive, may differ in the number of each type of machine. This implies that their originally designed product mixes may also differ. Cross-fab production is therefore needed to increase the total throughput of the two fabs.

3. Problem statement

This section describes the dual-fab route planning problem more precisely. We first present the assumptions that confine the context of the route planning problem, and then proceed to introduce the decision variables, objective function and constraints of the problem. In explaining the assumptions, the two fabs are respectively called *Fab_A* and *Fab_B*.

Assumption 1: *Each fab is functionally comprehensive*. Each of the two fabs is so comprehensively equipped that it can handle the manufacture of each product by itself – not requiring the functional support of the other fab.

Assumption 2: *A product has four possible routes*. To implement cross-fab production, the manufacturing route of a product is split into two parts, where the route's break point is called the *cut-off point*. The two parts can be manufactured in different fabs, and yield

two possible routes for cross-fab production. One, represented by $\alpha \rightarrow \beta$, denotes that the first part of the route is manufactured at *Fab_A* and the second part at *Fab_B*. The other part, represented by $\beta \rightarrow \alpha$, denotes that the first part of the route is at *Fab_B* and the second part at *Fab_A*. Since each fab is functionally comprehensive, a product thus has four possible manufacturing routes, α, β, $\alpha \rightarrow \beta$, and $\beta \rightarrow \alpha$, where α denotes the route at *Fab_A* only and β denotes the route at *Fab_B* only.

Assumption 3: *The transportation path between any two workstations/buffers is unique, rather than multiple*. In each fab, a transportation system for moving wafer jobs has been established. Theoretically, there may exist multiple paths to transport a wafer job from one workstation to another; however, to reduce the complexity of traffic control, we predefine a fixed path for such transport.

The route planning problem has two decision variables for each product: its *cut-off point* and the ratios of its four possible routes (simply called *route ratios*). Let the cut-off point and route ratios of product i be represented by $(\pi_i, \overline{r_i})$. Here, π_i denotes the identification code (an integer) of the operation for separating a route into two parts, and $\overline{r_i} = [a_i, b_i, c_i, d_i]$ is a four-element vector where each element denotes the percentage of a particular route – of the four routes α, β, $\alpha \rightarrow \beta$, and $\beta \rightarrow \alpha$. Define $\Pi = [\pi_1, \ldots, \pi_n]$ as a set of cut-off points and $R = [\overline{r_1}, \ldots, \overline{r_n}]$ as a set of route ratios for n products to be produced. The route planning problem is to determine a (π^*, R^*) in order to maximise the total throughput of the two fabs, subject to the constraint of meeting a target cycle time.

4. Solution framework

The framework proposed for solving the dual-fab route planning problem is shown in Figure 1, and involves two modules. In Module 1, each transportation path is assumed to be equipped with infinite capacity, and the transportation time between any two workstations/buffers is zero. With the routing problem so simplified, we attempt to find an optimum Π, in terms of minimising the total number of inter-fab transportations. The problem is solved by an iterative use of a linear program (LP) model. For a particular Π, the LP model computes the minimum number of inter-fab transportations, which is regarded as the performance of Π. We then use a binary search algorithm to identify an optimum Π^* as the ultimate decision for the cut-off point.

In Module 2, with the obtained Π^* taken as parameters, we deal only with the decision variables $R = [\overline{r_1}, \ldots, \overline{r_n}]$. In this module, each transportation path is taken as a tool with limited capacity. The transportation time required to pass along a path can be varied,

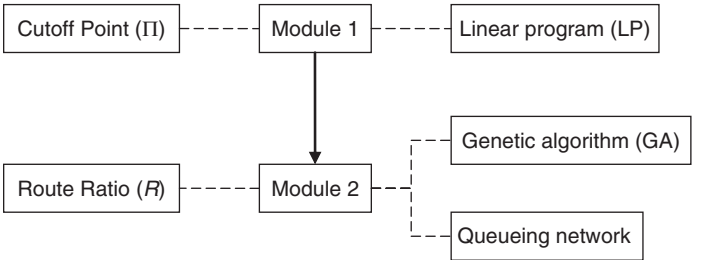

Figure 1. Solution framework.

depending upon the traffic flow intensity. The higher the traffic intensity, the longer the cycle time.

Module 2 involves two sub-modules. The first develops a performance evaluator for a particular (Π, R). To do so, we first construct a queueing network model (Connors *et al.* 1996) in order to compute the resulting mean cycle time, subject to a target throughput and a particular (Π, R). The queueing model is further enhanced as follows. Subject to a target mean cycle time and a particular (Π, R), the enhanced model can compute the resulting throughput – the performance of (Π, R).

With Π^* having been obtained in Module 1, the second sub-module of Module 2 searches for an R^* so that the performance (Π^*, R^*) is the best. A genetic algorithm is proposed to solve the search problem – finding the ultimate decision for R.

In summary, the solution space of the dual-fab route planning problem can be described by $S = \{(\Pi, R^*) \mid \Pi \in \Pi\_\ Set,\ R \in R\_\ Set\}$. The objective is to find an optimum (Π^*, R^*) from S, in terms of maximising the throughput subject to a target cycle time. Since the number of elements in S can be very large, the problem is decomposed into two sub-problems. The first is to find an optimum Π^*, and the second proceeds to find an optimum R^* by taking Π^* as predefined parameters.

The essence of these two modules is compared below. Module 1 essentially deals with a *static capacity allocation* problem that does not consider the job flow time. In contrast, Module 2 deals with a *time-phased capacity allocation* problem, in which the job flow time is addressed and computed by a queueing network model.

Without addressing the *job flow time*, Module 1 need not consider the transportation times of the jobs. This leads to the underlying assumption of Module 1 – the transportation time between any two workstations/buffers is zero. When the underlying assumption is released, we have to consider the job flow time in Module 1. Solving such a problem is very computationally expensive because it may need an iterative evaluation of a linear program embodied within a discrete event simulation program, as proposed by Hung and Leachman (1996).

5. Module 1: LP model and search algorithm

The solution for Module 1 is obtained by an iterative use of an LP program. We first describe the LP model and then present the iterative method – a bi-section search algorithm.

Indices

i	product index
g	workstation index in *Fab_A*
h	workstation index in *Fab_B*

Parameters

n	total number of products
π_i	cut-off point for defining the cross-fab routes of product i
Π	$\Pi = [\pi_i]$, $1 \le i \le n$, a vector for describing the cut-off points of all products
Q	estimated total throughput of the two fabs when in high utilization (in lots), which is used as the target throughput in the LP model
P_i	percentage of product i in the product mix, $\sum_{i=1}^{n} P_i = 1,\ 0 \le P_i \le 1$

C_g available machine hours of workstation g in *Fab_A*
C_h available machine hours of workstation h in *Fab_B*
m_a total number of workstations in *Fab_A*
m_b total number of workstations in *Fab_B*
W^a_{ig} total processing time per lot required on workstation g in *Fab_A*, when product i is manufactured by route α
W^c_{ig} total processing time per lot required on workstation g in *Fab_A*, when product i is manufactured by route $\alpha \rightarrow \beta$
W^d_{ig} total processing time per lot required on workstation g in *Fab_A*, when product i is manufactured by route $\beta \rightarrow \alpha$
W^b_{ih} total processing time per lot required on workstation h in *Fab_B*, when product i is manufactured by route β
W^c_{ih} total processing time per lot required on workstation h in *Fab_B*, when product i is manufactured by route $\alpha \rightarrow \beta$
W^d_{ih} total processing time per lot required on workstation h in *Fab_B*, when product i is manufactured by route $\beta \rightarrow \alpha$

Decision variables

a_i percentage use of route α in producing product i
b_i percentage use of route β in producing product i
c_i percentage use of route $\alpha \rightarrow \beta$ in producing product i
d_i percentage use of route $\beta \rightarrow \alpha$ in producing product i

5.1 *The LP model*

The LP program computes a minimum number of cross-fab transportations for a particular Π– a decision for the route cut-off points, which is known before solving the LP problem. The objective function of the LP program is denoted by $Z(\Pi)$:

$$\text{Min } Z(\Pi) = \sum_{i=1}^{n} Q \cdot P_i \cdot (c_i + d_i),$$

s. t.

$$a_i + b_i + c_i + d_i = 1, \quad 1 \le i \le n, \tag{1}$$

$$\sum_{i=1}^{n} Q \cdot P_i \cdot (a_i \cdot W^a_{ig} + d_i \cdot W^d_{ig} + c_i \cdot W^c_{ig}) \le C_g, \quad 1 \le g \le m_a, \tag{2}$$

$$\sum_{i=1}^{n} Q \cdot P_i \cdot (b_i \cdot W^b_{ih} + d_i \cdot W^d_{ih} + c_i \cdot W^c_{ih}) \le C_h, \quad 1 \le h \le m_b. \tag{3}$$

The objective function is to minimise the number of cross-fab production lots. The rationale for defining this objective is that cross-fab production requires a longer transportation time than within-fab production. Subject to a target cycle time, an attempt to minimise cross-fab production lots tends to increase the total throughput. Constraint (1) describes the dependent relationship among the route ratios. Constraints (2) and (3)

ensure that the capacity used in each workstation, for *Fab_A* and *Fab_B*, is less than its available supply.

5.2 *Bi-section search algorithm*

The bi-section search algorithm finds an optimum solution Π* from a space, denoted by {Π}, which is the possible combinations of cut-off points for all products. The algorithm is an iterative process. In an iteration, each product has only two possible cut-off points to select. Taking a product route as a line, the two cut-off points are, respectively, in the first and the third quartiles (Figure 2). By evenly cutting the route into two segments, each cut-off point is in the middle of a particular segment. Of the two evenly divided segments, the one where a cut-off point remains is called the *housing-segment* of the point.

In each iteration i, the size of the space {Π} is 2^n if there are n products. By solving the LP program in an exhaustive manner (i.e. 2^n times), we can obtain the best solution in this iteration – denoted Π_i^*, which defines the optimum set of cut-off points. For each product, the *housing-segment* of the cut-off point obtained is called the γ-segment (i.e. the remaining segment) of the product, which is the output of iteration i and will be the input of iteration $i+1$. The bi-section search algorithm is summarised below.

Algorithm: *Search_Cut-off_Points*

Initialisation

- For each product, take the whole route as its γ-segment.

For $i=1$ to N

- Create the two cut-off points on the γ-segment for each product
- Solve LP programs in an exhaustive manner to find Π_i^*
- Compute the γ-segment for each product based on Π_i^*

End for

Output the cut-off points for each product.

6. Module 2 – queueing and GA

The problem to be solved in Module 2 can be stated as follows. Given a target cycle time (CT_0) and a cut-off point decision (Π*) obtained from Module 1, we attempt to find an

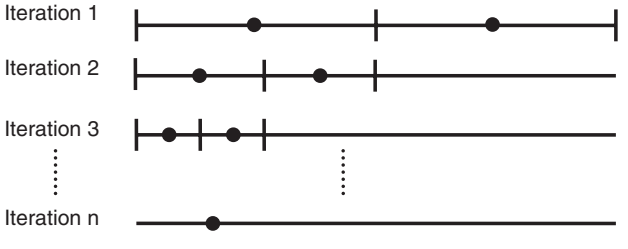

Figure 2. Process for obtaining the cut-off point.

optimal route ratio decision $R = [\overline{r_1}, \ldots, \overline{r_n}]$ in order to maximise the total throughput of the two fabs subject to the fact that the corresponding average cycle time is less than CT_0.

This problem is essentially a space search problem, with solution space $H = \{R\} = \{[\overline{r_1}, \ldots, \overline{r}_n] | \overline{r_i} = (a_i, b_i, c_i, d_i)\}$. A genetic algorithm is proposed to solve the problem. In the algorithm, the fitness (performance) of a solution R is evaluated by a queueing network model. We first introduce the queueing network model and proceed to the genetic algorithm.

6.1 *Queueing network*

The queueing network model is an extension of the model developed by Connors *et al.* (1996), the I/O function of which can be briefly formulated as follows: $CT = f(TH, R, \Pi)$. That is, given a target total throughput (TH), a route ratio decision (R), and a cut-off point decision (Π), the queueing model (f) can be used to compute the two fabs' mean cycle time (CT). However, Connors *et al.* (1996) did not consider the effect of transportation among workstations.

We extend the application of their model based on two assumptions. First, we assume that the transportation path between any two stations is unique, where a station is either a workstation or a WIP storage buffer. Secondly, each transportation path between any two stations is modeled as a 'conveyor machine' with only one unit of capacity. Such an extension makes the developed queueing model closer to a semiconductor fab in the real world. Likewise, the I/O function of the extended queuing model can also be described as $CT = f(TH, R, \Pi)$.

The objective function in Module 2 is to maximise the throughput (TH) subject to a target cycle time (CT_0). To evaluate the objective function, we used a *bi-section search technique* to find the total throughput (TH) for a particular route ratio (R); that is, $TH = f(R, *, CT_0) TH = f(R, *, CT_0)$, where * denotes the cut-off point decision obtained in Module 1 and CT_0 is the target cycle time. Note that, for the function $CT = f(TH, R, *)$, the larger the TH value, then the larger the CT value. The bi-section search technique, based on $CT = f(TH, R, *)$, searches for a value for TH so that $CT = CT_0$. The bi-section search algorithm is just like that for the binary search for a particular point on a line segment.

6.2 *Genetic algorithm*

The genetic algorithm (GA) identifies an optimal solution R^* from the space $\{R\}$. As stated, the performance of R is obtained from the enhanced queueing model. A possible solution R (called a chromosome) is represented by a vector $R = [\overline{r_1}, \ldots, \overline{r}_n]$, where $\overline{r_i} = (a_i, b_i, c_i, d_i)$. We call $\overline{r_i}$ a *gene-segment* and each of its elements a *gene*, and the gene values are by the constraints $a_i + b_i + c_i + d_i = 1$ and $0 \leq a_i, b_i, c_i, d_i \leq 1$.

The GA is an iterative algorithm and can be briefly described as follows.

Procedure GA

Step 1: Initialisation

- $t = 0$, Status = 'Not-terminate'
- Randomly generate N_p valid chromosomes to form a population P_0.

Step 2: Genetic Search

While (Status = 'Not-Terminate') do

- Use *cross-over* operator to create N_c new chromosomes
- Use *mutation* operator to create N_m new chromosomes
- Form a pool by taking the union of P_t and the set of newly created chromosomes
- $t = t + 1$, and select the best N_p chromosomes from the pool to form P_t
- Check if *termination condition* is met; if yes, set Status = 'Terminate'

Endwhile

Step 3: Step 3: Output the best chromosome R^* in P_t

The crossover operation creates two new chromosomes (say R_3 and R_4) from two existing ones (say R_1 and R_2). Let each *gene-segment* i in R_1 and R_2 be represented, respectively, by $\overline{r_{i1}}$ and $\overline{r_{i2}}$. We propose a one-point crossover operation (Binh and Lan 2007) on gene-segments $\overline{r_{i1}}$ and $\overline{r_{i2}}$ to create two new ones $\overline{r_{i3}}$ and $\overline{r_{i4}}$, which in turn could yield two new chromosomes $R_3 = [\overline{r_{i3}}]$ and $R_4 = [\overline{r_{i4}}]$, $1 \le i \le n$.

The one-point crossover operation on a gene-segment is briefly introduced. For two gene-segments (i.e. $\overline{r_{i1}}$ and $\overline{r_{i2}}$), we randomly choose a gene, swap their gene values, and modify the other gene value in order to ensure constraint compliance. Consider an example where the second gene is chosen as the cross-over point for mixing $\overline{r_{i1}} = (a_{i1}, b_{i1}, c_{i1}, d_{i1})$ and $\overline{r_{i2}} = (a_{i2}, b_{i2}, c_{i2}, d_{i2})$. By the swap and modification operations, we would obtain $\overline{r_{i3}} = (a_{i1}, b_{i2}, c_{i1}, 1 - a_{i1} - b_{i2} - c_{i1})$ and $\overline{r_{i4}} = (a_{i2}, b_{i1}, c_{i2}, 1 - a_{i2} - b_{i1} - c_{i2})$.

In the mutation operation, a new chromosome (say R_2) is created by an existing chromosome (say R_1). The mutation algorithm creates R_2 by modifying a particular gene-segment of R_1. The modified gene-segment is chosen randomly. While being selected, two of its genes are randomly chosen and their gene values are swapped. For example, if gene-segment i^* is chosen for modification, and the second and fourth genes are chosen to swap for $\overline{r_{i^*1}} = (a_{i1}, b_{i1}, c_{i1}, d_{i1})$, then $\overline{r_{i^*2}} = (a_{i1}, d_{i1}, c_{i1}, b_{i1})$, which in turn yields a new chromosome $R_2 = [\overline{r_{11}}, \ldots, \overline{r_{i^*2}}, \ldots, \overline{r}_{n1}]$ from $R_1 = [\overline{r_{11}}, \ldots, \overline{r_{i^*1}}, \ldots, \overline{r}_{n1}]$.

Two termination conditions are defined for the GA. First, the best solution in P_t is no change for a certain period (say T_b iterations). Second, population P_t has evolved over a certain number of iterations, that is t has reached its predefined upper bound (T_u).

7. Experiments

7.1 *Benchmarks and data*

Using numeric experiments, we attempt to evaluate the effectiveness of the proposed method. Two other methods are used as benchmarks for comparison. The proposed method is designated *LP-GA*, where *LP* denotes the linear program and *GA* denotes the genetic algorithm. The two benchmark methods are special cases of *LP-GA*. The first is called *M-GA*, which denotes that the cut-off point of each route has been predetermined – in the *middle* of the route. The second is called *N-GA*, which denotes that cross-fab production is *not* allowed. Such a comparison tells us how much benefit a dual-fab would obtain if the *LP-GA* method was used.

In the dual-fab experiments, the data for machines and product routes are adapted from an HP-fab from the literature (Wein 1988). Of the two fabs, one involves 93 machines

Table 1. Cut-off points obtained by the LP-GA program.

Total number of steps:	Product 1 172	Product 2 172	Product 3 150
RA	85th step	85th step	129th step
RB	84th step	84th step	78th step

Table 2. A comparison of the mean cycle times of different algorithms.

	R_A ($Q_A = 128$ lots)		R_B ($Q_B = 169$ lots)	
Algorithm	CT (min)	Gap (%)	CT (min)	Gap (%)
LP-GA	11,080	0	11,639	0
M-GA	12,175	9.88	12,811	10.06
N-GA	12,463	12.48	14,075	20.9

and the other 72 machines. Being functionally identical, each fab involves four batch workstations and 21 series workstations. The MTBF (mean time between failure) and MTTR (mean time to repair) of each machine is available, exponentially distributed. Three types of products are produced. One product involves 150 operations and the other two both involve 172 operations, but are different in processing times. In implementing the GA, we set $T_b = 1000$, $T_u = 30$, $P_0 = 100$, $P_{cr} = 0.8$, and $P_m = 0.1$.

7.2 *Performance comparison*

The three methods are compared in two scenarios, with product mixes $R_A = (3:2:5)$ and $R_B = (5:4:1)$. For each product mix, from the queueing model we obtain a throughput level that will keep the two fabs in high utilisation: $Q_A = 128$ lots and $Q_B = 169$ lots.

We compare the three methods from two perspectives. First, given a target throughput level, the mean cycle time of each method is compared. In the comparison, Q_A and Q_B are used as the target throughput levels. Second, given a target cycle time, we compare the throughput of each method. In the comparison, we set $CT_0 = 11{,}081$ min for R_A and $CT_0 = 11{,}445$ min for R_B.

The cut-off points of each route obtained by the *LP-GA* method are shown in Table 1, which indicates that the cut-off points suggested by the *LP-GA* are different from that of *M-GA*.

Table 2 shows a comparison of the mean cycle times, subject to a target throughput. The *LP-GA* outperforms the two benchmark methods. Using the result of *LP-GA* as a baseline, the cycle time of the *LP-GA* method is about 10% better than that of *M-GA*, and about 12–20% better than that of *N-GA*. This implies that managing a dual-fab by adopting an optimum cross-fab production policy tends to shorten the cycle time – significantly better than managing each fab independently (i.e. no cross-fab production).

Table 3 shows a comparison of the throughput, subject to a target cycle time. The *LP-GA* method also outperforms the two benchmark methods. Using the results for

Table 3. A comparison of the computing times for the throughput of different algorithms.

	R_A ($CT_0 = 11{,}081$ min)		R_B ($CT_0 = 11{,}445$ min)	
Algorithm	Throughput (lots)	Gap (%)	Throughput (lots)	Gap (%)
LP-GA	128	0	169	0
M-GA	125	2.34	165	2.37
N-GA	124	3.12	161	4.73

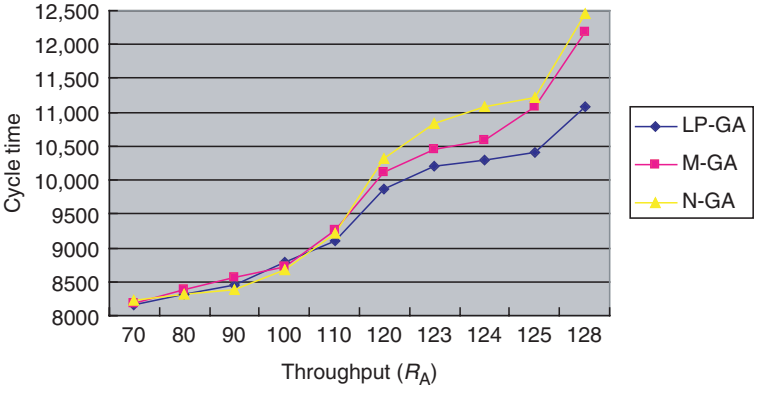

Figure 3. Relationship between throughput and cycle time for product mix R_A.

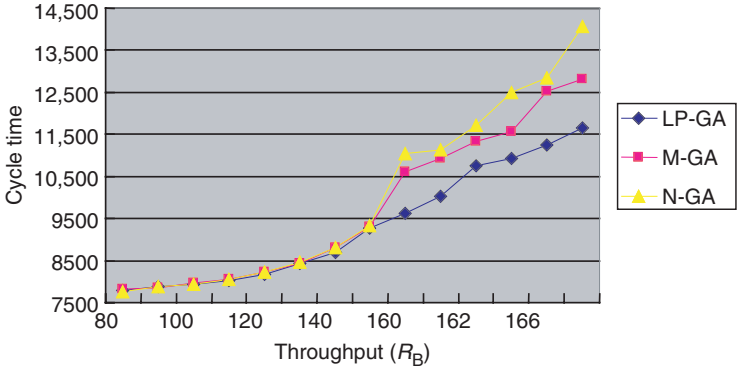

Figure 4. Relationship between throughput and cycle time for product mix R_B.

the *LP-GA* as a baseline, the throughput of the *LP-GA* method is about 2.3% higher than that of *M-GA*, and about 3.1–4.7% higher than that of *N-GA*. This implies that optimal planning of cross-fab production produces an increasing throughput.

Figures 3 and 4 reveal the relationship between cycle time and throughput for product mixes R_A and R_B, respectively. The higher the throughput, the longer the cycle time. The two figures also show that the higher the throughput, the larger the performance gap. That is, the contribution of the *LP-GA* method becomes greater when it is applied in a high-market-demand scenario.

Table 4. Computation times required by each module in the LP-GA method.

Route ratio	Module 1 (s)	Module 2 (s)
RA	3.5	95.578
RB	4.2	103.265

7.3 *Complexity analysis*

The computation times required by each module in the *LP-GA* method are shown in Table 4. The table indicates that the computation requirement of Module 2 is much greater than that of Module 1. Each of the two modules essentially deals with a space-search problem – attempting to find an optimal solution from a solution space. Module 1 adopts an analytic approach (a linear program), whereas Module 2 adopts a meta-heuristic approach (GA). A complexity analysis for Module 1 is therefore carried out below.

In Module 1, the iterative use of the linear program is based on a binary-search method. For a scenario with n products and each product involving $2^{x-1} < m \leq 2^x$ operations, the number of linear programs we have to perform is $N = x \cdot 2^n$. For scenarios with $n = 3$ and $m = 172$, we need to perform the linear program about $8 \cdot 2^3 = 64$ times, which computationally takes only about 4 s. The computation time will increase significantly if n is greatly increased.

To deal with scenarios with large n, future work in this direction is required. We need to develop a *product clustering* module. Of the n products, only a limited number (say c) are considered for cross-fab production, the remaining n–c products only being eligible for single-fab production.

8. Conclusion

This paper presents an approach to solving the route planning problem for a semiconductor dual-fab. In the problem, each product can be manufactured in either fab. Each product has four possible production routes, which are defined by a cut-off point. The route planning problem involves two decisions – determining the cut-off point and the route ratio for each product – in order to maximise the throughput subject to a cycle time constraint.

An *LP-GA* method is proposed to solve the route planning problem. We first use the LP module to make the cut-off point decisions, and proceed to use the GA module for making the decision concerning the route ratio. The *LP-GA* method is compared with two benchmark methods by numerical experiments. Results show that the *LP-GA* method significantly outperforms the other methods.

Extensions of this research are being considered. The first is the extension of this approach to a multiple-fab production system – for example, three or more fabs sharing production capacity. The second is the extension to a scenario with greater flexibility in production routes – for example, each product could have two or more cut-off points and in turn have more than four routes. The third extension as stated above is the examination of scenarios with a large number of products.

Acknowledgement

This research was supported financially by the National Science Council, Taiwan, under contract NSC-96-2628-E-009-026-MY3.

References

Avonts, L.H. and Wassenhove, L.N.V., 1988. The part mix and routing mix problem in FMS: a coupling between an LP model and a closed queueing network. *International Journal of Production Research*, 26 (12), 1891–1902.

Binh, Q.D. and Lan, P.N., 2007. Application of a genetic algorithm to the fuel reload optimization for a research reactor. *Applied Mathematics and Computation*, 187, 977–988.

Chiang, D., *et al*., 2007. Optimal supply chain configurations in semiconductor manufacturing. *International Journal of Production Research*, 45 (3), 631–651.

Connors, D.P., Feigin, G.E., and Yao, D.D., 1996. A queueing network model for semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 9 (3), 412–427.

Defersha, F.M. and Chen, M., 2006. Machine cell formation using a mathematical model and a genetic-algorithm-based heuristic. *International Journal of Production Research*, 44 (12), 2421–2444.

Dimopoulos, C., 2006. Multi-objective optimization of manufacturing cell design. *International Journal of Production Research*, 44 (22), 4855–4875.

Frederix, F., 2001. An extended enterprise planning methodology for the discrete manufacturing industry. *European Journal of Operational Research*, 129, 317–325.

Hung, Y.F. and Leachman, R.C., 1996. A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations. *IEEE Transaction on Semiconductor Manufacturing*, 9 (2), 257–269.

Karabuk, S. and Wu, S.D., 2003. Coordinating strategic capacity planning in the semiconductor industry. *Operations Research*, 51, 839–849.

Kim, C.O., Beak, J.G., and Jun, J., 2005. A machine cell formation algorithm for simultaneously minimizing machine workload imbalances and inter-cell part movements. *International Journal of Advanced Manufacture Technology*, 26, 268–275.

Lee, Y.H., *et al*., 2006. Supply chain model for the semiconductor industry in consideration of manufacturing characteristics. *Production Planning and Control*, 17 (5), 518–533.

Mahdavi, I., *et al*., 2006. A set partitioning based heuristic procedure for incremental cell formation with routing flexibility. *International Journal of Production Research*, 44 (24), 5343–5361.

ManMohan, S.S., 2005. Managing demand risk in tactical supply chain planning for a global consumer electronics company. *Production and Operations Management*, 14 (1), 69–79.

Mieghem, J.A., 1999. Coordinating investment, production, and subcontracting. *Management Science*, 45 (7), 954–971.

Nsakanda, A.L., Diaby, M., and Price, W.L., 2006. Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings. *European Journal of Operational Research*, 171, 1051–1070.

Rupp, T.M. and Ristic, M., 2000. Fine planning for supply chains in semiconductor manufacture. *Journal of Materials Processing Technology*, 107, 390–397.

Spiliopoulos, K. and Sofianopoulou, S., 2007. Manufacturing cell design with alternative routings in generalized group technology: reducing the complexity of the solution space. *International Journal of Production Research*, 45 (6), 1355–1367.

Toba, H., *et al*., 2005. Dynamic load balancing among multiple fabrication lines through estimation of minimum inter-operation time. *IEEE Transactions on Semiconductor Manufacturing*, 18 (1), 202–213.

Vin, E., Lit, P.D., and Delchambre, A., 2005. A multiple-objective grouping genetic algorithm for the cell formation problem with alternative routings. *Journal of Intelligent Manufacturing*, 16, 189–209.

Wein, L.M., 1988. Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing*, 1 (3), 115–130.

Wu, M.C. and Chang, W.J., 2007. A short-term capacity trading method for semiconductor fabs with partnership. *Expert Systems with Application*, 33 (2), 476–483.

Wu, S.D., Erkoc, M., and Karabuk, S., 2005. Managing capacity in the high-tech industry: a review of literature. *The Engineering Economist*, 50, 125–158.