Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 35 (2008) 938–945
Expert Systems with Applications
www.elsevier.com/locate/eswa

A multiple criteria decision for trading capacity between two semiconductor fabs

Muh-Cherng Wu *, Wen-Jen Chang [1]

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

Abstract

This paper presents a multiple criteria decision approach for trading weekly tool capacity between two semiconductor fabs. Due to the high-cost characteristics of tools, a semiconductor company with multiple fabs (factories) may weekly trade their tool capacities. That is, a lowly utilized workstation in one fab may sell capacity to its highly utilized counterpart in the other fab. Wu and Chang [Wu, M. C., & Chang, W. J. (2007). A short-term capacity trading method for semiconductor fabs with partnership. *Expert Systems with Application, 33*(2), 476–483] have proposed a method for making weekly trading decisions between two wafer fabs. Compared with no trading, their method could effectively increase the two fabs' throughput for a longer period such as 8 weeks. However, their trading decision-making is based on a single criterion—number of weekly produced operations, which may still leave a space for improving. We therefore proposed a multiple criteria trading decision approach in order to further increase the two fabs' throughput. The three decision criteria are: number of operations, number of layers, and number of wafers. This research developed a method to find an optimal weighting vector for the three criteria. The method firstly used NN + GA (neural network + genetic algorithm) to find an optimal trading decision in each week, and then used DOE + RSM (design of experiment + response surface method) to find an optimal weighting vector for a longer period, say 10 weeks. Experiments indicated that the multiple criteria approach indeed outperformed the previous method in terms the fabs' long-term throughput.

Keywords: Capacity trading; Semiconductor; Neural network; Genetic algorithm; Design of experiment; Response surface method

1. Introduction

The manufacturing of semiconductor products is a long-route process. A semiconductor product, called a *wafer*, involves hundreds of *operations*. These operations are generally grouped into tens of *layers*. The tools for semiconductor manufacturing are capital-intensive. A tool may cost up to ten millions of dollars. A typical wafer fab (a semiconductor factory) involves hundreds of tools, which as a whole may cost over 1 billion dollars. Therefore, effective management of tool capacity has been a significant research issue.

Traditionally, effective management of tool capacity has been investigated from three perspectives—in terms of decision horizon. From long-term perspective, some literature addressed the tool planning problem (Wu, Erkoc, & Karabuk, 2005); that is, how to purchase tools to fulfill a long-term forecasted demand that typically ranges from one to several years (Bard, Srinivasan, & Tirupati, 1999; Çatay, Erengüç, & Vakharia, 2003; Christie & Wu, 2002; Connors, Feigin, & Yao, 1996; Grewal, Bruska, Wulf, & Robinson, 1998; Hood, Bermon, & Barahona, 2003; Swaminathan, 2000, 2002; Wu, Hsiung, & Hsu, 2005). From medium-term perspective (ranging from one to several quarters), researchers addressed the product mix planning problems, which examined how to select customer orders to optimize the use of tool capacity (Bermon & Hood, 1999; Chou & Hong, 2000; Chung, Lee, & Pearn, 2003; Chung, Lee, & Pearn, 2005). From short-term perspective, the issues of

* Corresponding author. Tel.: +886 3 5731913; fax: +886 3 5720610.
E-mail addresses: mcwu@cc.nctu.edu.tw (M.-C. Wu), wjchang6@ms39.hinet.net (W.-J. Chang).
[1] Tel.: +886 5927700x2954; fax: +886 3 926848.

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.08.002

production planning (Odrey, Green, & Appello, 2001; Chen, Chen, Lin, & Rau, 2005) and shop floor control (Chen & Chen, 1996; Kuroda, Tomita, & Maeda, 1999; Mönch & Drießel, 2005) are addressed, which attempt to effectively use the tool capacity to produce the committed orders. Most literature based on the three perspectives essentially dealt with the tool capacity problem of a single fab.

A recent track on capacity planning turned to the mutual support of tools among different fabs. A semiconductor company usually has its fabs built in a cluster; that is, the fabs are close in their physical locations. This provides opportunities for tool capacity support among fabs. Deboo (2000) and Toba et al. (2005) took several different fabs as a single big one; a product before its completion may travel through more than one fab. They developed a dispatching algorithm for dynamically planning the route of a product. This approach implicitly assumed the adoption of a central management system. That is, each fab is not an autonomy unit; therefore, the performance of each individual fab may not be easily identified.

Managing each fab based on individual performance is undoubtedly important in motivating diligent work. A company may manage its multiple fabs by a de-centralized management system. That is, each fab operates in an autonomy manner and supports each other by trading tool capacity periodically. Wu and Chang (2007) proposed a method to find a weekly trading portfolio in order to maximize the two fabs' throughput in a longer period such as 8 weeks. They used total number of weekly completed operations as the criterion for finding the trading portfolio. Experiments indicated that such a *weekly* trading criterion indeed would increase the *longer-period* fab throughput.

However, our analysis indicates that the capacity trading criterion proposed by Wu and Chang (2007) still left a space for improving. Typically, a semiconductor product involves hundreds of operations, grouped into tens of layers. The productivity of a fab could be measured in three indices—number of operations, number of layers, and number of products (known as throughput). Intuitively, maximizing the output volume of operations would maximize that of layers, in turn that of products. However, a layer is a combination of specific operations; therefore, producing a higher number of operations may not lead to producing a higher number of layers if many operations are unable to be aggregated into a layer. This implies that the longer-period fabs' throughput could be possibly increased by adopting a new capacity trading criterion.

This paper proposes the use of a multiple criteria in the decision of weekly capacity trading between two fabs. Three trading criteria are considered: number of operations, number of layers, and number of products. To integrate these three criteria, a weighting for each has to be determined. Different weighting assignments lead to different trading outcomes, in turn different longer-period throughputs. We therefore attempt to identify an optimal weighting vector for the three trading criteria in order to maximize total profit of the two fabs of a longer period, such as 10 weeks.

The research framework involves two modules. The first module, for given a weighting vector of criteria, aims to determine its optimal weekly trading portfolio. That is, we attempt to find a *weekly* trading portfolio that would maximize $\alpha_1 \cdot O + \beta_1 \cdot L + \gamma_1 \cdot W$, where $[\alpha_1, \beta_1, \gamma_1]$ denotes the given weighs of criteria and O, L, and W respectively denotes the number of operations, layers, and wafers in that week. The second module attempts to find an optimal weighting vector $[\alpha^*, \beta^*, \gamma^*]$ in order to maximize the total profit of the two fabs—which would yield after the two fabs have traded capacity for a longer period, say 10 weeks. These two modules have been implemented and tested by numerical examples. Experiments indicated that the propose approach indeed outperform the previous work.

2. Module 1—finding weekly trading portfolio

The first module is to find a weekly trading portfolio that would maximize the fabs' weekly aggregated output. The aggregated output involves three components: number of operations, number of layers, and number of wafers, with the weights for integrating the three components being given. The development of this module involves three steps: (1) how to define the solution space of trading portfolios, (2) how to evaluate the performance of each trading portfolio, and (3) how to find a near-optimal solution.

2.1. Define solution space

In practice, a semiconductor fab comprises dozens of workstations; each workstation involves a number of functionally identical tools. For any two workstations in different fabs, we could trade their capacity if the two are functionally identical. Such a pair of workstations is called a *tradable pair* of workstations. Consider a case where the two fabs have m tradable pairs. The solution space of trading portfolios would include $N = \prod_{i=1}^{m}(B_i + 1)$ elements, where B_i denotes the maximum number of trading units for a tradable pair i; that is, its trade option could range from 0 to B_i units.

For a tradable pair i, we determine that $B_i = round\_up\left(\frac{1}{2u}|\rho_{s,i} - \rho_{b,i}| \times Q_{s,i} \times T\right)$, where u denotes the basic trading units; $\rho_{s,i}$ denotes the utilization of the workstation that sells capacity; $\rho_{b,i}$ denotes the utilization of the workstation that buys capacity; $Q_{s,i}$ denotes the number of tools in the workstation that sells capacity; and T denotes the number of working hours per week. Consider a case with $u = 20$ h, $\rho_{s,i} = 60\%$, $\rho_{b,i} = 90\%$, $Q_{s,i} = 5$ tools, and $T = 168$ h. Then $B_i = round\_up(6.3) = 7$ trading units. That is, the two workstations can trade at most 140 h.

Notice that the $\rho_{s,i}$ and $\rho_{b,i}$ refer to the workstation utilizations in the coming week while there is no capacity trading applied. The workstation utilizations are estimated by carrying out a discrete-event simulation. The simulation is a deterministic model; that is, the daily uptime of each tool is assumed to be a constant rather than stochastic. The adoption of this assumption is based on the simulation

findings by Kim, Shim, Choi, and Hwang (2003) whose experiment results advocated the use of deterministic simulation model in predicting a fab's short-term behavior such as one week. The deterministic simulation program is abbreviated *Det_Sim*(S_i, T_k), where S_i denotes the initial status of the two fabs at week i, and T_k denotes the trading portfolio applied at week i. In evaluating $\rho_{s,i}$ and $\rho_{b,i}$, we use *Det_Sim*(S_i,T_0), where T_0 denotes a no-trading decision.

2.2. Evaluate performance of trading portfolios

The performance of a trading portfolio is evaluated by the fabs' aggregated output $I = \alpha_1 \cdot O + \beta_1 \cdot L + \gamma_1 \cdot W$ where O, L, and W respectively denotes the total number of operations, layers, and wafers produced by the two fabs; and $[\alpha_1, \beta_1, \gamma_1]$ denotes the predefined weights for the three components. The higher the value of I, the more preferable is the trading portfolio.

As stated, the performance of a trading portfolio can be estimated by carrying out a deterministic simulation program. According to our experiments, one such estimation by using a typical personal computer (PC) takes about 40 s in computation. Consider a case where the fabs have four tradable workstations, each of which has at most 15 trading units. Then, the total number of trading portfolios would be $(15+1)^4 = 65{,}536$, requiring about 30 days.

This research proposed two methods to reduce the computation time. Firstly, the neural network (NN) technique is used to emulate the function of simulation, which could yield results in a much speedy way. Experiments indicated that, by the NN technique, the computation time per estimation would be much less than one second. Secondly, we used the genetic algorithm (GA) technique to reduce the number of estimations; that is, trading portfolios are not exhaustively evaluated—only a limited number of "seemingly good" trading portfolios are tested.

Consider the simulation program as a transformation mechanism. Then, the input is a trading portfolio and the output involves three components (O, L and W). The application of NN technique is to construct an input/output mapping to emulate the function of the simulation program. To do so, we firstly sample K trading portfolios and compute their performances by simulation. The obtained K pairs of input/output vectors are then used to construct a neural network for each fab by the back-propagation algorithm (Fausett, 1994; McClelland & Rumelhard, 1988; Rumelhart, Hinton, & Williams, 1986).

This algorithm would establish a non-linear mapping between the input/output vectors. That is, given an input vector, the neural network through the non-linear mapping could compute the output of each fab speedily. The neural network is called well-trained if its projected output vectors are close to those obtained from the simulation; typically their degree of discrepancies is measured by RMSE (root mean squared errors). A well-trained neural network could then be interpreted as a "faster simulator". By constructing a neural network for each fab, we could then speedily compute the aggregated output of the two fabs.

2.3. Find a near-optimal solution

With a well-trained neural network, we used the GA technique to find a near-optimal trading portfolio from the huge solution space. The GA technique has been widely used in solving a large space search problem and has been justified in its effectiveness in finding a near-optimal solution (Mitchell, 1998). The GA used in this research is briefly described below.

Each trading portfolio is modeled as a string of numerals, called *chromosome*. Each number in the string, called *gene*, represents a trading option for a particular type of workstation. The performance of a trading portfolio is called the *fitness* of the chromosome. Changing the gene values of a existing chromosome leads to the creation of a new one. Methods for such a chromosome creation are called genetic operators.

The solution search mechanism of the GA is by evolving a *population* of chromosomes, which is limited in number during the evolution. An evolution *generation* means that the population has been updated once. In the population updating process, some chromosomes are replaced by "seemingly good" new ones, which are created by genetic operators. The genetic operators involve three types: reproduction, crossover, and mutation. The population updating process terminates when either the best solution cannot be improved further or the population has evolved over a predefined number of generations. And the best chromosome in the final population is the trading portfolio provided by the GA.

2.4. An integrated procedure

The aforementioned steps of the first module can be summarized below by a procedure called *Find_Weekly_Trading*($[\alpha,\beta,\gamma]$,S_i), where $[\alpha,\beta,\gamma]$ is the given weighting vector and S_i denotes the initial status of the two fabs at week i—the particular week for making the trading decision. The output of the procedure is denoted by $Q_i^*(\alpha,\beta,\gamma)$, which represents the optimal trading portfolio at week i.

Procedure *Find_Weekly_Trading*($[\alpha,\beta,\gamma]$,S_i)

Step 1: Use *Det_Sim*(S_i,T_0) to estimate the utilization of each tradable pair of workstations
Step 2: Define the solution space N of trading portfolios
Step 3: Establish a neural network to emulate *Det_Sim*
- Randomly sample K trading portfolios
- For each of the K trading portfolios, use *Det_Sim*(S_i,T_k) to compute its weekly outputs of O, L, W in each fab
- Construct a neural network for each fab
- Aggregate the projected outputs of the two fabs

Step 4: Use GA to search an optimal trading portfolio $Q_i^*(\alpha,\beta,\gamma)$ from solution space N.

3. Module 2—select optimum weights for trading criteria

The ultimate purpose of trading tool capacity between fabs is to make their long-term total profits as higher as possible. In the first module, we have presented how to find a weekly optimum capacity trading decision based on a particular weighting vector $[\alpha,\beta,\gamma]$. This implies that the use of different weighting vectors may yield different total profits. Therefore, how to find an optimum weighting vector is important.

The second module therefore attempts to find an optimal weighting vector $[\alpha^*,\beta^*,\gamma^*]$ for aggregating the three trading decision criteria—number of operations (O), number of layers (L), and number of wafers (W) in order to maximize the total profit of the two fabs, from a longer-period perspective (T weeks). That is, making trading decisions successively for $T=10$ weeks based on the vector $[\alpha^*,\beta^*,\gamma^*]$ would maximize the 10 weeks total profit of the two fabs.

This module involves two procedures. The first procedure is to compute the T weeks total profit of the two fab, while the weighting vector $[\alpha,\beta,\gamma]$ has been given. The second procedure is to find an optimum weighting vector $[\alpha^*,\beta^*,\gamma^*]$ from the set $S=\{[\alpha,\beta,\gamma]|\alpha\ +\beta+\gamma=1,\ 0\leqslant\alpha,\beta,\gamma\leqslant 1\}$.

3.1. Compute total profit in T weeks

The procedure to compute the T weeks total profit for a given weighting vector $[\alpha,\beta,\gamma]$, called *Compute_Profit* (α,β,γ), is described below with its notation firstly introduced.

Notation

$[\alpha,\beta,\gamma]$ the given weighting vector

$Q_i^*(\alpha,\beta,\gamma)=[w_1,w_2,\ldots,w_k]$ an optimum trading portfolio at week i, where w_i denotes the number of trading units for tradable workstation i and k denotes the number of tradable pairs

S_i the initial status of the two fabs at week i, which describes the WIPs before each workstation and the up-or-down status of each machine

F_i the final status of the two fabs at week i, which describes the WIPs before each workstation and the up-or-down status of each machine; i.e.,$F_i=S_{i+1}$

$V_i=[v_1,v_2,\ldots,v_m]$ total produced volumes of each product in the two fabs at week i, where m is the number of product types

$R_i=[r_1,r_2,\ldots,r_m]$ contribution margin of each product at week i, where m is the number of product types

P_i total profit of the two fabs at week i

$P(\alpha,\beta,\gamma)$ total profit of the two fabs in T weeks

Procedure *Compute_Profit*(α,β,γ)

Step 0: Initialization, input $[\alpha,\beta,\gamma]$ and S_1
For $i=1,\ldots,T$
Step 1: Use the procedure *Find_Weekly_Trading*($[\alpha,\beta,\gamma]$, S_i) to compute $Q_i^*(\alpha,\beta,\gamma)$
Step 2: Use a stochastic simulation program *Sto_Sim*(S_i, $Q_i^*(\alpha,\beta,\gamma)$) to estimate S_{i+1} and V_i after implementing the trading portfolio $Q_i^*(\alpha,\beta,\gamma)$
Step 3: Compute total profit $P_i=V_i\cdot R_i^T$ at week i

Endfor
Step 4: Compute the total profit of the T weeks $P(\alpha,\beta,\gamma)=\sum_{i=1}^{T}P_i$

In Step 2 of the above procedure, the stochastic simulation program *Sto_Sim*(S_i, $Q_i^*(\alpha,\beta,\gamma)$) is used for predicting the behavior of the two fabs after carrying out a capacity trading portfolio; that is, it takes $Q_i^*(\alpha,\beta,\gamma)$ and S_i as input and generates S_{i+1} and V_i as output, which could be used to obtain the total profit $P(\alpha,\beta,\gamma)$.

Notice that in evaluating the longer-period (say, $T=10$ weeks) performance of a trading portfolio, we have to use a stochastic simulation rather than a deterministic simulation. This is to reflect the unpredictable nature of a real wafer fab and attempts to justify the robustness of a weekly optimum trading portfolio.

3.2. Find an optimum weighting vector

We now discuss how to find an optimum weighting vector $[\alpha^*,\beta^*,\gamma^*]$ from the set $S=\{[\alpha,\beta,\gamma]|\alpha+\beta+\gamma=1,\ 0\leqslant\alpha,\beta,\gamma\leqslant 1\}$ in order to maximize the total profit of the T weeks. The techniques of mixture design of experiments (DOE) and response surface method (RSM) are used. That is, we sample a number of weighting vectors, compute their total profits, and use the obtained results to construct a response surface, which is polynomial function that could be used to quickly estimate the total profit for each $[\alpha,\beta,\gamma]$ in S.

Using an appropriate experiment design model is very important in effectively predicting the behavior of a system. In our experiment, the three factors are imposed by a constraint $\alpha+\beta+\gamma=1$ and are not independent. Therefore, the simplex centroid design (Montgomery, 1991) is used for the three mixture components. As shown in Table 1, 10 design points are selected in the experiment; the total profit $P(\alpha,\beta,\gamma)$ for each design point could be computed by applying the procedure *Compute_Profit*(α,β,γ). The 10 pairs of $[(\alpha,\beta,\gamma),P]$ are then used to construct the response surface. With the response surface, an optimum weighting vector $[\alpha^*,\beta^*,\gamma^*]$ could then be easily identified by analytical approaches such as the gradient method (Myers & Montgomery, 1995).

3.3. Rationales for choosing optimization techniques

Notice that in the first module we use the technique of NN + GA in finding an optimal weekly trading portfolio.

Table 1
Simplex centroid design points for a mixture experiment

Design point	α	β	γ
1	1	0	0
2	0	1	0
3	0	0	1
4	1/2	1/2	0
5	1/2	0	1/2
6	0	1/2	1/2
7	1/3	1/3	1/3
8	2/3	1/3	1/3
9	1/3	2/3	1/3
10	1/3	1/3	2/3

In contrast, in the second module, we use the technique of DOE + RSM to find an optimal weighting vector. A question may arise: why two different techniques are used in searching an optimum solution?

By its very nature, the technique of DOE + RSM requires much less number of sampled solutions (only 10 points in this application) in constructing the response surface. Therefore, it is more suitable for cases with a longer computation time per solution evaluation, whereas the technique of NN + GA is more suitable for cases with a shorter computation time per solution evaluation.

In this research, the second module attempts to evaluate the performance of a weighting vector $[\alpha, \beta, \gamma]$. That is, for a given weighting vector $[\alpha, \beta, \gamma]$, we need to compute the optimum trading portfolio for each of the T weeks. This would become quite time-consuming if we attempt to evaluate a large number of weight vectors $[\alpha, \beta, \gamma]$. To limit the number of sampled weighting vectors, the technique of DOE + RSM is thus used in the second module.

4. Numerical example

The proposed approach has been tested by a numerical example. Table 2 shows the number of product mixes, number of workstations, contribution margin of each product, and some other features of process routes in the two trading fabs, where a type of product is denoted by xPyM which specifies that the product has x poly-layers and y metal layers (Xiao, 2001). The contribution margin of each product and the process routes are disguised from those provided by industry. And we assumed that only four types of workstations are tradable between the two fabs.

The two simulation programs *Det_Sim* and *Sto_Sim* are coded by using eM-plant (Tecnomatix Technologies Ltd, 2001). The experiment design and response surface methods are carried out by using MINITAB (Minitab Inc., 2003). The neural network and the GA programs are coded by using C programming language.

In the example, the two fabs are designed to trade capacity from week 1 to week 10 (i.e., $T = 10$). The fabs' initial status (S_1) is generated by *Sto_Sim*, which starts from an empty fab until comes to a steady state by uniformly releasing wafers based on the given product mixes.

In constructing an NN for modeling a fab's behavior for a particular week, we sampled $K = 2000$ trading portfolios. Then we use *Det_Sim* to compute the weekly fab output of each trading portfolio. Subsequently, the input/output of the 2000 trading portfolios are used to construct a neural network for the week. By using a typical PC (personal computer), it takes about 40 s for carrying out the simulation once. We used 50 PCs in a lab to do the computation. That is, 40 simulation runs have to be performed on a PC and takes about 26.7 min. With the 2000 data being available, the computation time for constructing the neural network is much faster, about less 1.5 min by using only one PC. Likewise, the computation effort for the GA to obtain $Q_i^*(\alpha, \beta, \gamma)$ takes less than 1.5 min. With the $Q_i^*(\alpha, \beta, \gamma)$ being available, we used *Sto_Sim* to compute the initial fab status of week $i + 1$, which takes about 1.0 min in computation.

Therefore, the computation time for running simulations, constructing a neural network, carrying out the GA search, and updating the fabs' initial status of next week takes about 30 min in total. That is, with a computation facility of 50 PCs, the two fabs can make their weekly trading decision in 30 min, if the weighting vector of the decision criteria $[\alpha, \beta, \gamma]$ has been given. In practice, half an hour decision-making is acceptable for a weekly trading decision.

In contrast, the computation time for finding an optimal weighting vector $[\alpha^*, \beta^*, \gamma^*]$ is much more computationally extensive. For a given weighting vector $[\alpha, \beta, \gamma]$, we have to compute the optimal weekly trading decisions for 10 weeks, which will take 5 h (0.5 h/week * 10 weeks) computation on a computing facility with 50 PCs. In the DOE, we have 10 weighting vectors to be evaluated, and this will take about 50 h computation—seemingly an astonishing value! However, the decision of an optimal weighting vector $[\alpha^*, \beta^*, \gamma^*]$ is not made weekly. A practical application maybe updates $[\alpha^*, \beta^*, \gamma^*]$ quarterly, which then takes 50 h of computation—seemingly acceptable to a semiconductor fab. This computation time surely can be reduced if more number of PCs is used.

Table 2
Tools and products for the two fabs

FAB	Number of workstations	Total number of tools	Product	Total processing time (h)	Total number of operations	Contribution margin
Fab_A	60	275	4P1M	400	358	65
			1P7M	440	412	80
Fab_B	60	201	1P3M	318	276	50
			1P8M	480	446	100

Table 3
Ten weekly optimal trading portfolios for $[\alpha,\beta,,\gamma]=[1,0,0]$

Week	Trading portfolio $[\alpha,\beta,\gamma]=[1,0,0]$
1	(180, 240, 960, 320)
2	(180, 220, 780, 300)
3	(160, 240, 640, 280)
4	(160, 240, 600, 260)
5	(180, 220, 620, 300)
6	(140, 220, 560, 300)
7	(160, 200, 600, 300)
8	(180, 180, 640, 280)
9	(160, 200, 600, 300)
10	(160, 180, 680, 240)

Table 1 shows the 10 weighting vectors selected for constructing the response surface. Table 3 shows the weekly optimum trading portfolios for the 10 weeks while $[\alpha,\beta,\gamma]=[1,0,0]$. Table 4 shows total profit of the 10 weeks for each of the 10 weighting vectors in the DOE. The equation of the response surface with $R^2=0.852$ is shown below, where y denotes the total profit divided 1000; that is, in unit of thousand dollars. The contour and three-dimensional plots of the response are shown in Fig. 1. A flag in the contour plot identifies the region that maximizes the total profit of the two fabs

$$y = 209.53\alpha + 211.90\beta + 213.19\gamma + 29.64\alpha\beta + 17.34\beta\gamma + 25.32\alpha\gamma - 25.32\alpha\beta\gamma + 17.30\alpha^2\beta - 17.30\alpha\beta^2 + 59.25\alpha^2\gamma - 59.25\alpha\gamma^2$$

The embedded algorithm in MINITAB for searching an optimal solution from the response surface is then used to find $[\alpha^*,\beta^*,\gamma^*]=(0.63,0.13,0.24)$. As shown in Table 4, the total profit obtained by using $[\alpha^*,\beta^*,\gamma^*]$ exceeds those obtained by the 10 sampled weighting vectors. In addition, the table also indicates that the decision of no trading is the lowest in total profit. This confirmed the claim of Wu and Chang (2007); that is, weekly trading tool capacity is effective in generating more profit. Compared to Wu and Chang (2007), this research further increases the total profit by using an optimal weighting vector in weekly trading.

The extraordinary outcome of the proposed approach has two important implications. Firstly, it advocates the use of multiple criteria in trading weekly capacity between fabs. Secondly, a weekly performance evaluation of a

Table 4
The responses of 10 design points

Design point	Criteria weights			Throughput of each product				Total profit	Total output
	α	β	γ	4P1M	1P7M	1P3M	1P8M		
1	1	0	0	993	628	787	554	209,535	2962
2	0	1	0	1001	638	792	562	211,905	2993
3	0	0	1	1013	645	799	558	213,195	3015
4	1/2	1/2	0	1024	661	820	577	218,140	3082
5	1/2	0	1/2	1041	648	816	574	217,705	3079
6	0	1/2	1/2	1031	651	812	572	216,895	3066
7	1/3	1/3	1/3	1027	659	823	581	218,725	3090
8	2/3	1/3	1/3	1085	627	833	585	220,835	3130
9	1/3	2/3	1/3	1024	655	808	568	216,160	3055
10	1/3	1/3	2/3	1025	641	810	557	214,105	3033
No. trading	–	–	–	882	569	689	488	186,100	2628
Optimal trading	0.63	0.13	0.24	1088	675	841	591	225,870	3195

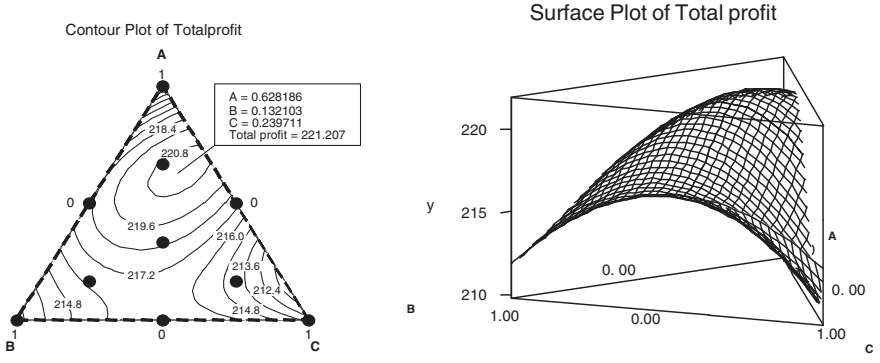

Fig. 1. Contour and 3-D plot for the response.

semiconductor fab may better be evaluated from three perspectives—number of operations, number of layers, number of wafers. Due to the long cycle time characteristics of semiconductor manufacturing, managers in practice tend to evaluate a fab's weekly performance by the number of produced operations, and evaluate its quarterly performance by the number of wafers produced. To align the short-term (week) and relative long-term (quarter) objectives, the weekly performance evaluation may need a multiple criteria such as proposed.

5. Concluding remarks

This paper presents a method for determining an optimal weighting vector $[\alpha^*, \beta^*, \gamma^*]$ for a multiple criteria decision on weekly trading tool capacity between two semiconductor fabs. The three decision criteria involve number of operations (O), number of layers (L) and number of wafers (W). That is, if the two fabs make weekly trading decision by maximizing the aggregated output $\alpha^* O + \beta^* L + \gamma^* W$, their resulting long-term profits will reach a maximum.

In the proposed method, we used the techniques of NN + GA to obtain an optimal trading portfolio for each week while the weighting vector $[\alpha, \beta, \gamma]$ is given. The data set for constructing a NN is generated by a deterministic simulation program *Det_Sim*. The total profit in the case of using a weighting vector $[\alpha, \beta, \gamma]$ is estimated by a stochastic simulation program *Sto_Sim*. Finally, the techniques of DOE + RSM are used to find an optimal weighting vector $[\alpha^*, \beta^*, \gamma^*]$, where a design point in the DOE denotes a sampled weight vector.

In terms of total profit, experiments indicated that the proposed multiple criteria approach outperformed the previous study that uses only a single criterion—number of operations. This finding has two important implications. Firstly, it advocates the use of multiple criteria in trading tool capacity. Secondly, a weekly performance evaluation of a semiconductor fab may better be evaluated from three perspectives—number of operations, number of layers, number of wafers. In practice, managers tend to evaluate a fab's weekly performance by the number of produced operations, and evaluate its quarterly performance by the number of wafers produced. To align the short-term (week) and long-term (quarter) objectives, the weekly performance evaluation may need a multiple criteria such as proposed.

Some possible enhancements of this research are being investigated. Firstly, how to reduce the computation efforts in evaluating the performance of a weighting vector, which now takes about 5 h on 1 facility with 50 PCs. Secondly, we attempt to justify whether the adoption of a smaller/longer time bucket for making capacity trading decisions would improve the total profits further. A smaller time bucket may denote that a trading decision is made every 3 days, while a longer time bucket may denote that a trading decision is made every 2 weeks.

References

Bard, J. F., Srinivasan, K., & Tirupati, D. (1999). An optimization approach to capacity expansion in semiconductor manufacturing facilities. *International Journal of Production Research, 37*(15), 3359–3382.

Bermon, S., & Hood, S. J. (1999). Capacity optimization planning system. *Interfaces, 29*(5), 31–50.

Çatay, B., Erengüç, Ş. S., & Vakharia, A. J. (2003). Tool capacity planning in semiconductor manufacturing. *Computers & Operations Research, 30*, 1349–1366.

Chen, J. C., Chen, C. W., Lin, C. J., & Rau, H. (2005). Capacity planning with capability for multiple semiconductor manufacturing fabs. *Computers & Industrial Engineering, 48*, 709–732.

Chen, L. H., & Chen, Y. H. (1996). A design procedure for a robust job shop manufacturing system under a constraint using computer simulation experiments. *Computers & Industrial Engineering, 30*(1), 1–12.

Chou, Y. C., & Hong, I. H. (2000). A methodology for product mix planning in semiconductor foundry manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 13*(3), 278–285.

Christie, R. M. E., & Wu, S. D. (2002). Semiconductor capacity planning: Stochastic model and computational studies. *IIE Transaction, 34*, 131–143.

Chung, S. H., Lee, A. H. I., & Pearn, W. L. (2003). Product mix optimization for semiconductor manufacturing based on AHP and ANP analysis. *The International Journal of Advanced Manufacturing Technology, 25*(11–12), 1144–1156.

Chung, S. H., Lee, A. H. I., & Pearn, W. L. (2005). Analytic network process (ANP) approach for product mix planning in semiconductor fabricator. *International Journal of Production Economics, 96*, 15–36.

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

Deboo, S. (2000). Block cross processing: An innovative approach to constrain management. *The ninth international symposium on semiconductor manufacturing* (pp. 225–228).

Fausett, L. (1994). *Fundamental of neural networks: Architectures, algorithms, and applications*. Prentice Hall.

Grewal, N. S., Bruska, A. C., Wulf, T. M., & Robinson, J. K. (1998). Integrating targeted cycle-time reduction into the capital planning process. In *Proceedings of the 1998 winter simulation conference, Washington, DC* (pp. 1005–1010).

Hood, S. J., Bermon, S., & Barahona, F. (2003). Capacity planning under demand uncertainty for semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 16*(2), 273–280.

Kim, Y. D., Shim, S. O., Choi, B., & Hwang, H. (2003). Simplification methods for accelerating simulation-based real-time scheduling in a semiconductor wafer fabrication facility. *IEEE Transactions on Semiconductor Manufacturing, 16*(2), 290–298.

Kuroda, M., Tomita, T., & Maeda, K. (1999). Dynamic control of a cellular-line production system under variations in the product mix. *International Journal of Production Economics, 60–61*, 439–445.

McClelland, J. L., & Rumelhard, D. E. (1988). *Explorations in parallel distributed processing*. Cambridge, MA: MIT.

Minitab Inc. (2003). *MINITAB manual*.

Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT Press.

Mönch, L., & Drießel, R. (2005). A distributed shifting bottleneck heuristic for complex job shops. *Computers & Industrial Engineering, 49*, 363–380.

Montgomery, D. C. (1991). *Design and analysis of experiments*. New York: Wiley.

Myers, R. H., & Montgomery, D. C. (1995). *Response surface methodology*. New York: Wiley.

Odrey, N. G., Green, J. D., & Appello, A. (2001). A generalized Petri net modeling approach for the control of re-entrant flow semiconductor wafer fabrication. *Robotics and Computer Integrated Manufacturing, 17*, 5–11.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating error. *Nature, 323*, 533–536.

Swaminathan, J. M. (2000). Tool capacity planning for semiconductor fabrication facilities under demand uncertainty. *European Journal of Operational Research, 120*, 545–558.

Swaminathan, J. M. (2002). Tool procurement planning for wafer fabrication facilities: A scenario-based approach. *IIE Transaction, 34*, 145–155.

Tecnomatix Technologies Ltd. (2001). *EM-PLANT objects manual*. Germany: Tecnomatix Software Company.

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

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

Wu, M. C., Hsiung, Y. I., & Hsu, H. M. (2005). A tool planning approach considering cycle time constraints and demand uncertainty. *International Journal of Advanced Manufacturing Technology, 26*(5), 565–572.

Wu, S. D., Erkoc, M., & Karabuk, S. (2005). Managing capacity in the high-tech industry: A review of literature. *The Engineering Economist, 50*(2), 125–158.

Xiao, H. (2001). *Introduction to semiconductor manufacturing technology*. Prentice Hall.