行政院國家科學委員會專題研究計畫　期中進度報告

子計畫三:結合模擬與排序佳化法的生產排程及其應用於 12 吋晶圓製造之研究(2/3)

計畫類別： 整合型計畫
計畫編號： NSC93-2213-E-002-043-
執行期間： 93 年 08 月 01 日至 94 年 07 月 31 日
執行單位： 國立臺灣大學電機工程學系暨研究所

計畫主持人： 張時中

計畫參與人員： 林偉誠、何元祥、于宗源、賴加將、廖祐楷、廖柏鈞、陳俊宏教授、謝博偉博士

報告類型： 精簡報告
報告附件： 出席國際會議研究心得報告及發表論文
處理方式： 本計畫可公開查詢

中　華　民　國 94 年 6 月 5 日

結合模擬與排序佳化法的生產排程及其應用於12 吋晶圓製造之研究(2/3)

Research on Simulation-based Ordinal Optimization Methods with Applications to Production Scheduling of 300mm Foundry Fabs (2/3)

計 畫 編 號： NSC93-2213-E-002-043
主 持 人：張時中教授
計畫參與人員： 林偉誠、何元祥、于宗源、賴加將、廖祐楷、
廖柏鈞、陳俊宏教授、謝博偉博士

執行期限：93 年8 月1 日至94 年7 月31 日
執行單位：國立臺灣大學電機工程學系暨研究所

中文摘要

本計畫三年的目標：(1) 為排序佳化模擬設計削減搜尋空間的新方法，(2) 將方法開發成工具模組，作為整合系統佳化平台的一部份，(3) 將模擬進行排序佳化應用在次世代半導體廠的有效生產排程。第一年中我們以平穩(Stationary)馬可夫決策問題為載具，提出模擬排序策略疊代(OOBPI) 的構想，利用策略疊代的架構，以模擬排序佳化來估算每一狀態的cost-to-go 函數值與最佳決策。初步模擬結果顯示這個構想的計算效能較傳統模擬為基礎的策略疊代法可快十倍。

第二年中我們完成OOBPI演算法的設計，包括方法理論、收斂分析、數值驗證及推展至一般動態規化的初探。在半導體晶圓生產模擬器方面，我們加入產品優先序之考量，以模擬半導體廠具有不同優先次序之生產排程。目前正結合OOBPI方法與半導體晶圓生產模擬器進行生產排程組合選擇的研究。

關鍵詞: 排序佳化、策略疊代、演算法設計、生產模擬、產品優先序、生產排程

Abstract

The three-year research objectives of this project are (1) to design search space reduction methods for simulation-based ordinal optimization (OO), (2) to develop these methods into tool modules as part of an integrated system optimization platform, and (3) to apply simulation-based OO to effective production scheduling of 300mm foundry fabs. In the first year, we considered the class of Stationary Markov decision problems as the conveyer problem. We proposed an idea of OO-Based Policy Iteration (OOBPI) to handle the combinatorial complexity of decisions over the time axis. Utilizing the framework of policy iteration, we approximate the optimal cost-to-go and optimal decision of each state by simulation-based OO. Preliminary numerical studies indicated one order of speed-up of OOBPI over the traditional simulation-based policy iteration.

In the second year, we have completed the design of OOBPI algorithm, including the theories for the method, convergence analysis, simulation study and exploration of possible extension to general dynamic programming. In developing a simulator of semiconductor wafer fabrication with differentiated services, we have incorporated priority service discipline into the simulator. We are now combining OOBPI with the fab simulator to study dynamic composition of production schedules for fabs.

***Key words*:** *Ordinal Optimization, Policy Iteration, Algorithm design, Fab Simulator, Priority, Production Scheduling*

二、計畫緣由與目的

Ordinal optimization (OO) is an emerging approach for efficient optimization of complex systems. Instead of finding the optimal solution, ordinal optimization selects a good enough solution from a subset of options from the design space based on a certain criteria and a specified confidence level. The principle investigator (PI) of this proposal has conducted research on applying simulation-based OO and optimal computing budget allocation (OCBA) to scheduling semiconductor wafer fabrication. Motivated by the findings of previous research and based on the foundation established, the PI considers simulation-based OO of good potential for good enough and efficient production scheduling/dispatching of next generation fabs, where the problem complexity and the need for optimization are much higher than those of current fabs.

The PI has been is performing a research under NSC support on search space reduction by combining design of experiment (DOE) and simulation-based OO (OODOE). In this three-year research project, the PI will adopt the salient production scheduling/dispatching problems of 300mm foundry fab as the conveyers and continue to advance his research in both **design of new methods** and **applications to production**. Specific objectives are
(1) to design search space reduction method for simulation-based OO, (2) to develop these methods into tool modules as part of an integrated system optimization platform, and to apply simulation-based OO to effective production scheduling of 300mm foundry fabs.

In the first year, we considered the class of Stationary Markov decision problems as the conveyer problem. We proposed an idea of OO-Based Policy Iteration (OOBPI) to handle the combinatorial complexity of decisions over the time axis. Utilizing the framework of policy iteration, we approximate the optimal cost-to-go and optimal decision of each state by simulation-based OO. Preliminary numerical studies indicated one order of speed-up of OOBPI over the traditional simulation-based policy iteration.

1. 第二年目的

In the second year, our objectives have been on

(1) the complete design of OOBPI algorithm, including the theories for the method, convergence analysis, simulation study and exploration of possible extension to general dynamic programming, and
(2) the development of a simulator of semiconductor wafer fabrication with differentiated services,
(3) combination of OOBPI with the fab simulator to study dynamic composition of production schedules for fabs.

三、研究方法

Part I: Design and Analysis of the OOBPI Algorithm

2. Simulation-based Policy Iteration (SBPI)

Markov Decision Problem (MDP) is one class of stochastic optimal control problems [1,2,3]. A MDP is characterized by states, controls, state transition probabilities, transition costs, and decision horizon. In this paper, we consider a stationary MDP with finite controls and states. To formulate the problem, let us define some notations first.

Notations:

α : Discount factor, $\alpha \in [0,1]$;

S : Set of states;

I : State index, $i \in S$;

U : Set of controls;

u : Control index, $u \in U$;

$U(i)$: Set of admissible controls at state i, $U(i) \subseteq U$;

T : Decision Time horizon;

π : Policy, a function mapping form state space to control space, $S \rightarrow C$;

t : Index of time epoch, when a state transition occurs, t=1,…,T;

u_t : Control at time t, and $u_t = \pi(i_t)$;

i_t : State at time t;

ω_t : Randomness outcome at time t with $\omega_t \in \Omega$, $\forall t$;

$f(\cdot,\cdot,\cdot)$: State transition function, $S \times U \times \Omega_s \rightarrow S$;

S_i : Set of states that can be reached one transition from state i;

$\{i_t\}$: A realization of stat trajectory i_t, $t = 1,2,\mathrm{K}$;

$p_{ij}(u)$: Transition probability from state i to state j under control u;

$g(i,u,j,\omega_t)$: transition cost from state i to state j under control u with an outcome ω_t

In the MDP, the transition cost function $g(i,j,\omega)$ and the transition probability $p_{ij}(u)$ have to be estimated via Monte-Carlo simulation. Given a stationary policy π, the discounted expected cost-to-go (CTG) of an initial state i is defined as

$$J_\pi(i) = \lim_{T\to\infty} E_w\left\{\sum_{t=0}^{T-1} \alpha^t \cdot g(i_t, i_{t+1}, \omega_t) \mid i_0 = i\right\} \quad (2.1)$$

The state transition is described by

$$i_{t+1} = f(i_t, \pi(i_t), \omega_t),\ \mathrm{t}= 0,1,2,\ldots, \quad (2.2)$$

where ω_t's are i.i.d. The stationary MDP (SMDP) of this paper is then

$$\min_\pi J_\pi(i_0) \text{ subject to (2.2) for any } i_0 \in S \quad (2.3)$$

2.1 SBPI algorithm

In the literature, Cooper et. al. [4] have proposed a simulation-based policy iteration (SBPI) algorithm to solve a stationary MDP problem with minimizing the average transition cost as the objective. Their SBPI algorithm is described as follows.

SBPI consists of two steps: policy evaluation and policy improvement. The policy evaluation step performs a one-step analysis that estimates the effect of each admissible control $u \in U(i_0)$ by using the control u for $t = 0$ while assuming a given policy π for $t \geq 1$. Given a stationary policy π and an initial state i, the policy iteration step first runs Monte-Carlo simulation to obtain N sample paths of length T. The cost-to-go under policy π can be estimated by simulation as

$$\hat{J}_\pi(\boldsymbol{\omega} \mid i = i_1, T-1) \equiv \sum_{t=1}^{T-1} \alpha^t \cdot g(i_t, u_t, i_{t+1}, \omega_t) \quad (2.4)$$

where $\boldsymbol{\omega} = \{\omega_t, t=1,\ldots,T\text{-}1\}$ and $\{i_t, t = 1,...,T-1\}$ is a sample state trajectory generated according to state equation (2.2). Then define a one-step CTG (1-CTG) as

$$\hat{r}_\pi(\boldsymbol{\omega} \mid i,u,T) \equiv \sum_{j\in S_i} p_{ij}(u) \cdot [g(i,u,j,\omega_0) + \alpha \cdot \hat{J}_\pi(\boldsymbol{\omega} \mid j,T)] \quad (2.5)$$

and $p_{ij}(u)$ is either known a priori or estimated from simulation. The time horizon parameter T will be omitted in our later discussions for notational simplicity.

Let the sample mean and variance of the1-CTG be $\bar{E}_\omega[\hat{r}_\pi(\boldsymbol{\omega} \mid i,u)]$ and $\bar{\sigma}^2_\omega[\hat{r}_\pi(\boldsymbol{\omega} \mid i,u)]$ respectively. The 95% confidence interval of the sample mean [12] can be approximated by $\left[\bar{E}_\omega(\hat{r}_\pi(\boldsymbol{\omega} \mid i,u)) - 1.96\frac{\bar{\sigma}^2_\omega[\hat{r}_\pi(\boldsymbol{\omega} \mid i,u)]}{\sqrt{N}},\right.$

$\left.\bar{E}_\omega(\hat{r}_\pi(\boldsymbol{\omega} \mid i,u)) + 1.96\frac{\bar{\sigma}^2_\omega[\hat{r}_\pi(\boldsymbol{\omega} \mid i,u)]}{\sqrt{N}}\right]$.

Define a simulation accuracy index

$$\beta \equiv 1.96 \frac{\overline{\sigma}^2_{\omega}\left[\hat{r}_{\pi}(\boldsymbol{\omega} \mid i,u)\right]}{\sqrt{N}} / \bar{E}_{\omega}\left(\hat{r}_{\pi}(\boldsymbol{\omega} \mid i,u)\right) \times 100\%, \quad (2.6)$$

which implies that the simulation is stopped when the half-length of the 95% confidence interval over the center point is less than β. It is intuitively clear that the larger the N, the smaller the β and the higher accuracy of $\bar{E}_{\omega}\left[\hat{r}_{\pi}(\boldsymbol{\omega} \mid i,u)\right]$.

The policy improvement step updates the policy by selecting for each state the best control among all admissible controls of the state based on their sample means of 1-CTG. The policy evaluation and policy improvement steps iterate till the policy stays the same from one iteration to another.

SBPI Algorithm

Step (s1). Initialization

Input an initial policy π^0, a simulation accuracy index β^*, initial number of simulation replications N_0, and an incremental number of simulation replications τ;

$k = 0$;

Step (s2). Policy Evaluation

For every state $i \in S$

For every admissible control $u \in U(i)$

2.1 Set $N = N_0$ and $\pi = \pi^k$;

2.2 Simulate $\hat{r}_{\pi}(\boldsymbol{\omega} \mid i,u)$ for N replications, each with T time epochs, and compute $\bar{E}_{\omega}\left[\hat{r}_{\pi}(\boldsymbol{\omega} \mid i,u)\right]$ and $\overline{\sigma}^2_{\omega}\left[\hat{r}_{\pi}(\boldsymbol{\omega} \mid i,u)\right]$;

2.3 If $1.96 \cdot \frac{\overline{\sigma}^2_{\omega}\left[\hat{r}_{\pi}(\boldsymbol{\omega} \mid i,u)\right]}{\sqrt{N}} \Big/ \bar{E}_{\omega}\left[\hat{r}_{\pi}(\boldsymbol{\omega} \mid i,u)\right] \le \beta^*$, go to step (s3);

2.4 Simulate additional τ replications;

$N = N + \tau$, go to step 2.2;

Step (s3). Policy Improvement

For every state $i \in S$

$$\pi^{k+1}(i) \in \arg\min_{u \in U(i)} \left\{\bar{E}_{\omega}\left[\hat{r}_{\pi}(\omega \mid i,u)\right]\right\}.$$

Step (s4). Iteration

If $\pi^{k+1} = \pi^k$, then output π^k and stop.

Otherwise, set $k = k + 1$ and go to step (s2)

The main challenge of SBPI is that with Monte-Carlo simulation, $\bar{E}_{\omega}\left(\hat{r}_{\pi}(\boldsymbol{\omega} \mid i,u)\right)$ converges slowly at rate $O\left(1/\sqrt{N}\right)$ [6, 7]. The total simulation time can be prohibitively expensive if we want to have an accurate estimation of CTG in policy evaluation. However, it is unclear in SBPI how the accuracy of 1-CTG estimation affects the identification of the best admissible control of each state, i.e., how it affects the policy improvement.

2.2 Simulation Study of SBPI

To investigate the relationship between the estimation accuracy (EA) of CTG and the accuracy of identifying the best control of each state, i.e., policy accuracy (PA), we perform a simulation study of SBPI. We design a study example of discounted stationary MDP with 4 states, 10 admissible controls for every state and α=0.9, where the state transition probabilities are randomly generated. Let us first define PA of SBPI.

Definition 2.1 Policy accuracy (PA) of SBPI

Consider applying SBPI to run over N replications of the example MDP. Let $\hat{\pi}^*_n$ be the control policy obtained by SBPI from the nth replication and π^* the optimal control policy of the MDP. Define a score function

$$I_n(i) = \begin{cases} 1, & \text{if } \hat{\pi}^*_n(i) = \pi^*(i), \text{ for every } i \in S; \\ 0, & \text{otherwise.} \end{cases} \quad (2.7)$$

The PA of SBPI is then

$$PA \equiv \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{|S|}\sum_{i \in S} I_n(i)\right]. \quad (2.8)$$

We then run SBPI over 100 replications of the example MDP and vary the EA requirement, i.e., β^*. Fig 2.1 shows how PA varies with β^*, where PA degrades with the increase of β^* in an approximate rate of 0.68. Fig 2.2 shows the CPU time needed to obtain the required EA, β^*, of 1-CTG. Note that the CPU time decreases rapidly with the increase of β^*. For example, when β^* is relaxed from 1% to 7%, the PA decreases from 99% to 84% while the CPU time needed decreases from 38 to 10 seconds. Such results indicate that relaxing EA requirement for 1-CTG estimation by simulation still leads to reasonably good PA but may largely reduces the computation time.

3. Ordinal Optimization-based Policy Iteration (OOBPI) Algorithm

The simulation study of SBPI in Section 2 indicates that a good policy accuracy can be achieved at relatively loose accuracy of the 1-CTG estimation by simulation. A bit sacrifice of PA may lead to significant reduction of simulation time in estimating the 1-CTG of each admissible control for a given state. The identification of the best control for each state essentially ranks individual controls in an ascending order of the 1-CTG estimates. Such observations motivate our combination of ordinal optimization [5] with SBPI into a new and efficient method for solving SMDP.

By exploiting the ordinal optimization concept, the key design idea of the OOBPI algorithm lies in replacing the EA requirement in the policy evaluation step of SBPI with a lower bound criterion on the probability of correct ranking among controls. That the top-ranking control obtained by simulation for a state is actually the best control for the state is defined as the event of correct selection (CS). In each iteration of OOBPI, the simulation for 1-CTG estimation over all the admissible controls of a state stops once a good enough control is identified, i.e., probability of CS $\equiv P\{CS\} > p^*$, where p^* is a pre-specified probability. Theory of ordinal optimization says that $P\{CS\}$ converges exponentially with respect to the number of simulation replications, N, while the EA of SBPI converges slowly at rate $O\left(1/\sqrt{N}\right)$ [6, 7]. So, adoption of OO in each iteration may lead to a good enough policy improvement at a

moderate p^* requirement. The good enough policy improvement in turn may get the estimated 1-CTG closer to the optimal CTG in the next iteration. Computation efficiency may thus be expected.

In specific, the OOBPI design applies the theoretic results of OO as follows.

Theorem 1 *Exponential Convergence of P{CS}*
Suppose the simulation replications for each control are i.i.d. and the simulation replications between any two admissible controls are independent. Assume that $\hat{r}_\pi(\omega \mid i,u)$ has a finite moment generating function. Then *P{CS}* converges to one exponentially. More specifically, there are $\alpha > 0$ and $\beta > 0$ such that $P\{CS\} \geq 1 - \alpha e^{-\beta N}$.
Proof: Theorem 5.1 of [8].

While the *confidence probability P{CS}* could converge at an exponential rate, a critical issue in applying it to *ordinal optimization* is the estimation of the *P{CS}* itself. Using a Bayesian model, [13] developed an lower bounding technique to *P{CS}* when the number of designs is large.

Theorem 2 *Lower Bound Approximation of P{CS}*
Let $\bar{E}_\omega\left(\hat{r}_\pi(\omega \mid i,u)\right)$, $u \in S_i$, denote the random variable whose probability distribution is the posterior distribution of the expected performance for control u under a Bayesian model. A lower bound to *P{CS}* for a state i can be calculated as,

$$P_i\{CS\} \geq \prod_{u \in U(i), u \neq b} P\{\bar{E}_\omega\left(\hat{r}_\pi(\omega \mid i,b)\right) < \bar{E}_\omega\left(\hat{r}_\pi(\omega \mid i,u)\right)\} \quad \text{绑}$$

≡*Approximate Probability of Correct Selection for state* i (*APCS(i)*), (3.1)

where b is the control with minimum $\bar{E}_\omega\left(\hat{r}_\pi(\omega \mid i,u)\right)$ among all $u \in S_i$.
Note that the computation of *APCS* is simply a product of pairwise comparison probabilities, which is simple to compute. Assume that $g(i,u,j,\omega_t)$ be independent among all state pairs (i,j) and be i.i.d over time with mean and variance $\left(\mu_{ij}, \sigma_{ij}^2\right)$. Then according to the central limit theorem, $\hat{J}_\pi(i,T)$ in Eq. (2.4) and the 1-CTG $\hat{r}_\pi(\omega \mid i,u)$ are can be approximated by random variables of normal distributions.

Let $\hat{r}_\pi(\omega \mid i,u) \sim \mathrm{N}\left(\bar{E}_\omega\left(\hat{r}_\pi(\omega \mid i,u)\right), \bar{\sigma}_\omega^2\left[\hat{r}_\pi(\omega \mid i,u)\right]\right)$,

the pairwise comparison probability in *APCS(i)*

$$P\{\bar{E}_\omega\left(\hat{r}_\pi(\omega \mid i,b)\right) < \bar{E}_\omega\left(\hat{r}_\pi(\omega \mid i,u)\right)\}$$

$$= \Phi\left(\frac{\bar{E}_\omega\left[\hat{r}_\pi(\omega \mid i,u)\right] - \bar{E}_\omega\left[\hat{r}_\pi(\omega \mid i,b)\right]}{\sqrt{\frac{\bar{\sigma}_\omega^2\left[\hat{r}_\pi(\omega \mid i,u)\right]}{N} + \frac{\bar{\sigma}_\omega^2\left[r_\pi(\omega \mid i,b)\right]}{N}}}\right), \quad (3.2)$$

where Φ is the standard normal accumulative distribution.
APCS for a state i is therefore

$$APCS(i) = \prod_{u \in U(i), u \neq b} P\{\bar{E}_\omega\left(\hat{r}_\pi(\omega \mid i,b)\right) < \bar{E}_\omega\left(\hat{r}_\pi(\omega \mid i,u)\right)\}$$

$$= \prod_{u \in U(i), u \neq b} \Phi\left(\frac{\bar{E}_\omega\left[\hat{r}_\pi(\omega \mid i,u)\right] - \bar{E}_\omega\left[\hat{r}_\pi(\omega \mid i,b)\right]}{\sqrt{\frac{\bar{\sigma}_\omega^2\left[\hat{r}_\pi(\omega \mid i,u)\right]}{N} + \frac{\bar{\sigma}_\omega^2\left[\hat{r}_\pi(w \mid i,b)\right]}{N}}}\right) \quad (3.3)$$

According to [13], *APCS* should provide a good approximation to $P\{\mathrm{CS}\}$. It is therefore adopted in OOBPI for calculating a lower-bound of the probability that the top-ranking control by simulation is indeed the optimal control for the state.

The detailed OOBPI algorithm is summarized as follows:

OOBPI Algorithm

Step O1: initialization.
Input an initial policy π^0, a simulation confidence level of correct selection P^*, and an incremental number of simulation replications τ;
k = 0;

Step O2: Policy evaluation
Set $N = N_0$, and 打冻 π^k;
O2.1 For every state $i \in S$
For every admissible control $u \in U(i) \subset U$
O2.2 Simulate $\hat{r}_\pi(\omega \mid i,u)$ for more replications, each with T time epochs;
Compute $\bar{E}_\omega\left[\hat{r}_\pi(\omega \mid i,u)\right]$ and $\bar{\sigma}_\omega^2\left[\hat{r}_\pi(\omega \mid i,u)\right]$;

Step O3: Policy improvement
O3.1 Find the optimal control b based on estimated 1-CTG,
i.e. $b \equiv \arg\left\{\underset{u \in U(i)}{Min}\, \bar{E}_\omega\left[\hat{r}_\pi(\omega \mid i,u)\right]\right\}$;
O3.2 Calculate $APCS(i)$ according to (3.3);
O3.3 Check if $APCS(i) > P^*$ for all i, then
$\pi^{k+1}(i) = b$ and stop,
Else set $N = N + \tau$,
go to Step O2.1;

Step O4: Iteration
If $\pi^{k+1} = \pi^k$, then stop and set $\pi^* = \pi^k$;
otherwise $k = k+1$ and return to Step O2.

The major differences of OOBPI with respect to SBPI lie in steps O3.2 and O3.3. The primary objective of OOBPI is to identify good admissible controls that perform relatively better than others rather than to accurately estimate the cost-to-go for all controls under consideration.

3. Convergence Analysis of OOBPI

In this section, we shall provide easily-verifiable sufficient conditions under which the OOBPI algorithm converges asymptotically to an optimal policy. The sufficient conditions are derived by utilizing the convergence analysis of SBPI [4] and a $P\{\mathrm{CS}\}$ approximation technique in the ordinal optimization context [14]. There are four steps in the derivation:

i) conversion of SBPI for an MDP problem with a discounted objective function to SBPI for an equivalent MDP problem with an objective of average reward;
ii) derivation of sufficient conditions on number of simulation replications under which SBPI converges in application to the MDP problem with an objective of average reward by exploiting the result of [10];
iii) establishment of the relationship between a lower bound of $P\{\text{CS}\}$ and the number of simulation replications per iteration;
iv) derivation of $P\{\text{CS}\}$ requirements under which OOBPI converges asymptotically.

Detailed analysis and derivation are as follows.

i) *Conversion of SBPI for discounted to average reward*

In a stationary MDP, the average reward for state i under policy π is defined as

$$v_\pi(i) = \lim_{T\to\infty} \frac{1}{T} E_\omega \left\{ \sum_{t=0}^{T-1} g(i_t, \pi(i_t), i_{t+1}, \omega_t) \mid i_0 = i \right\} \tag{4.1}$$

The discounted reward in (2.1) and the average reward then have the following relationship:

$$J_\pi(i) = \frac{1}{1-\alpha} v_\pi(i), \ \forall i \in S \tag{4.2}$$

It can be shown that by replacing $\hat{r}_\pi(\mathbf{\omega} \mid i,u)$ in (2.5) with $(1\text{-}\alpha)\hat{r}_\pi(\mathbf{\omega} \mid i,u)$, the SBPI finds the optimal policy for the corresponding MDP with average reward.

ii) *Sufficient convergence conditions of SBPI*

In the convergence analysis of BPI for SMDP problems with an objective of average reward, Cooper et al. provided a sufficient convergence of SBPI [4].

Proposition 1 (Proposition 5.5 of [4])

Let N_k, k=1, 2,..., be the number of simulation replications in iteration k of SBPI for SMDP problems with an objective of average reward. If

$$\sum_{k=0}^{\infty} \frac{1}{N_k} < \infty,$$

then the SBPI converges to an optimal policy with probability one.

Proposition 1 states that the simulation replications per iteration must grow "fast enough" as SBPI iterates to converge to the optimal policy. One simple example of such sequences is $\{N_k\} = \{n_0^k\}$, where n_0 is an integer and $n_0 > 1$. In this case, $\sum_{k=0}^{\infty} \frac{1}{N_k} = \frac{n_0}{n_0 - 1} < \infty$.

iii) *Relationship between number of simulation replications and probability of correct selection*

We now establish the relationship between the number of simulation replications N_k and $P_i\{\text{CS}\}$ of the best control for state i based on the sample mean and variance of the 1-CTG, $\bar{E}_\omega\left[\hat{r}_\pi(\mathbf{\omega} \mid i,u)\right]$ and $\bar{\sigma}^2_\omega\left[\hat{r}_\pi(\mathbf{\omega} \mid i,u)\right]$. From [14], *APCS* in Eq. (3.3) asymptotically approaches

$$\prod_{u\in U(i), u\neq b} \left(1 - \exp(\lambda(\pi,i) N_k)\right) \tag{4.3}$$

where $\lambda(\pi,i) = \dfrac{\delta^2_{b,u}(\pi,i)}{2\left(\sigma^2_b(\pi,i) + \sigma^2_u(\pi,i)\right)}$, and $\delta_{b,u}(\pi,i) = \bar{E}_w\left[\hat{r}_\pi(w \mid i,u)\right] - \bar{E}_\omega\left[\hat{r}_\pi(\omega \mid i,b)\right]$.

Let P_k^* be the simulation confidence level of correct selection for iteration k, 轩be an upper bound to $\lambda(\pi,i)$ for all π and I, and $P_k^* = \left(1 - e^{-\nu N_k}\right)^{|s|-1}$. In one iteration of OOBPI, P_k^* should be set so that at least N_k replications are run for each state i to achieve

$$APCS(i) \geq P_k^* = \left(1 - e^{-\nu N_k}\right)^{|s|-1} \tag{4.4}$$

That is

$$N_k = \frac{1}{-\nu} \ln\left(1 - (P_k^*)^{1/(|s|-1)}\right). \tag{4.5}$$

iv) *Sufficient convergence conditions of OOBPI*

It is desirable that sufficient condition setting of $\{P_k^*\}$ for OOBPI convergence does not require a prior statistics of the MDP. To ensure that $\frac{N_{k+1}}{N_k} = n_0$, $\{P_k^*\}$ is determined by applying Eq. (4.4) and we have

$$\frac{N_{k+1}}{N_k} = \frac{\frac{1}{-\nu}\ln\left(1 - (P_k^*)^{1/(|s|-1)}\right)}{\frac{1}{-\nu}\ln\left(1 - (P_{k+1}^*)^{1/(|s|-1)}\right)} = n_0. \tag{4.5}$$

Thus,

$$P_{k+1}^* = \left(1 - \left(1 - (P_k^*)^{1/(|s|-1)}\right)\right)^{|s|-1}. \tag{4.6}$$

Theorem 3: Sufficient Convergence Conditions of OOBPI

Let P_k^* be the confidence level of correct selection for iteration k, $0 < P_k^* < 1$ and $\{P_k^*\}$ satisfies Eq. (4.6). Then OOBPI converges to the optimal policy with probability one.

Proof: To achieve P_k^* by OOBPI at iteration k, it requires at least N_k simulation replication for each state i and $\{N_k\}$ satisfies the sufficient convergence conditions of SBPI.

5. Simulation Study

Now we perform simulation study of OOBPI over two stationary MDP examples. The first example is the same as that in Subsection 2.2 and the second example is a replacement problem taken from [15], which has 40 states, 41 admissible controls per state and the discount factor 脉凑伎俩 Relationship among the setting of *P{CS}* for estimation of 1-CTG by simulation, policy accuracy and computation time is the key of this study.

Although SBPI and OOBPI are different in the accuracy setting for 1-CTG estimation by simulation, simulation results of the two algorithms are still put in referential contrast. We set 刍冻ıfor SBPI to get the optimal policy for

each example.

In the evaluation of SBPI, various α values are used but it stays a constant among iterations in one run. We set $P_k^* = (65 + 5 \cdot k)\%$ for OOBPI, where P_k^* tightens up the accuracy as the iteration goes.

In the simulation study over example 1, within the policy accuracy (PA) range of 85%~98%, the corresponding P_k^* ranges from 70%~85% as depicted in Fig 5.1. The CPU time of SBPI increases from 4.2 seconds to 12.4 seconds in this PA range (Fig 2.2) while the corresponding CPU time range of OOBPI is 1.1 secs ~4.2 secs (Fig 5.2).

In the simulation study over example 2, Fig 5.3 shows the CPU time of each iteration of SBPI and OOBPI. The last item in Fig 5.3 is the total CPU time to reach an optimal policy. We find CPU time of each iteration in SBPI is almost the same. This is because we select a fixed β. The CPU time of OOBPI increases as it iterates. This is because P_k^* increases as iteration increases. Note that in this study, P_0^* can be very loose at the initial iteration but still leads to convergence to the optimal policy. By comparing the performance of OOBPI over examples 1 and 2, we can also observe that as the MDP size increases, OOBPI may have good advantage in both efficiency and optimality.

6. Fab Simulator with Priority Queues

We enhance the Fab simulator developed by Hu and Chang [21] with priority queueing disciplines. Two fab models are considered: FAB1 and FAB 2.

6.1 FAB1 and FAB2 Models

FAB1 is adopted from an aggregated full-scale production line previously studied by [17]. This model consists of 60 production stages, 12 different machine groups and a total of 40 machines. There is only one part type. Its sequence of processing steps and the machine group used by each step are given in Figure 6.1. Machines are subject to random failures, and each failed machine requires a random repair time. All the times to failure, times to repair, and processing times have exponential distributions with various values of mean time to failure (MTTF), mean time to repair (MTTR) and mean processing time (MPT).

FAB2 closely follows the model of [18], which is based on real life data and models the production facility of a semiconductor company. This model is represented by 13 different machine groups with jobs of 10 product families and the job class mix ratio is set as γ_i = 0.1 for i = 1, …,10. The last process steps of all the products will finish at MG 13. Its sequence of processing steps and the MG used by each step are given in Table 6.1. Table 6.2 lists the basic information of the model. Note that different machine groups have different processing time distributions in this FAB2 model.

6.2 Addition of Priority Queue to Simulator

The simulator in existence can simulate the situation for multiple part types manufactured by different machine groups, and consider deterministic re-entrant process flows. But in practice, each part types maybe has different priority to manufacture. Addition of priority queue to simulator is divided into two parts, one is single node study, and the other is fab queueing network study, as Figure 6.2.

In the first step, we add priority into the system which has single server and two priority part types called M/M/1 non-preemptive priority queue. We compare the simulation results with the analysis results [19], and observe the covariance between priorities. The arrival and service process of each types are not always exponential distribution in real world, so we will simulate the general situation, and replace M/M/1: PR queue by GI/G/1: PR queue.

In the research of GI/G/1: PR queue, we surveyed many papers and find out the analysis results from [20]. In that paper, the priority queue is converted to as many separated G/G/1-Special Service queues as many different priority levels are, shown in Figure 6.3. And it distinguishes the system time into three components: remaining service time of current customer under service, busy period of the higher priority queues and service time of the given customer. The moments of the busy period are:

$$E(R_\alpha(C_{A_\alpha}(S_j))) = \frac{\rho_\alpha}{1-\rho_\alpha}\tau_i \tag{6.1}$$

$$E(R_\alpha(N_\alpha)) = E(N_\alpha)\cdot\frac{\tau_\alpha}{1-\rho_\alpha} \tag{6.2}$$

$$E(R_\alpha^2(C_{A_\alpha}(S_j))) = \frac{\tau_j\tau_\alpha\rho_\alpha(c_{S_\alpha}^2 + c_{A_\alpha}^2)}{(1-\rho_\alpha)^3} + \frac{\rho_\alpha^2\tau_j^2(1+c_{S_j}^2)}{(1-\rho_\alpha)^2} \tag{6.3}$$

$$E(R_\alpha^2(N_\alpha)) = E(N_\alpha)\cdot\frac{\tau_\alpha}{(1-\rho_\alpha)^2} + E(N_\alpha^2)\cdot\frac{\tau_\alpha^2(c_{S_\alpha}^2 + \rho_\alpha c_{A_\alpha}^2)}{(1-\rho_\alpha)^3} \tag{6.4}$$

The next step after single node study, we will add priority into our existent fab queueing network simulator with re-entrant process flows, and observe the performance between each type with different priorities, then verify its simulation results.

四、結論與成果

In the second year of research, ordinal optimization (OO) has been combined with simulation-based policy iteration (PI) into an OOBPI algorithm. The OOBPI algorithm exploits simulation observations that a good enough selection among admissible decisions of each state needs only a relatively rough estimation of the cost-to-go function (CTG) values by simulation. Convergence of OOBPI to optimality can be achieved by properly setting, at each iteration, the requirement for probability of correctly selecting the best controls for individual states. Simulation studies have demonstrated the potential of OOBPI in computation efficiency and optimality. In developing a simulator of semiconductor wafer fabrication with differentiated services, we have incorporated priority service discipline into the simulator.

On-going research efforts are:

(1) exploration of possible extension of OOBPI to general dynamic programming,
(2) combination of OOBPI with the fab simulator to study dynamic composition of production schedules for fabs,
(3) development of an empirical fab behavior model that describes how fab capacity allocation and priority scheduling affects fab performance and variability.

Publications supported by this project

1. T.-K. Hwang, S-C. Chang, W.-L. Jan, "Properties of Iterative Proportional Capacity Allocation for Re-entrant Line Operations," *Proceedings of APIEMS*, Gold Coast, Australia, Dec. 13-15, 2004.
2. S.-C. Chang, C. H. Chen, M.-C. Chang, Y.-H. Ho, "Design of Ordinal Optimization-based Policy Iteration," submitted to *IEEE Conference on Decision and Control*, Dec. 2005.
3. W.-C. Lin, S.-C. Chang, "Hybrid Algorithms for Satellite Imaging Scheduling," submitted to *IEEE Transactions on Systems, Man and Cybernetics*, May 2005.

五、參考文獻

[1] M. L. Puterman, *Markov Decision Process*, John Willy & Sons, Inc., NY, 1994.
[2] D. Bertsekas, and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athenas, 1996.
[3] Y. He, Michael C. Fu, and Steven I. Marcus, "A Simulation Based-Policy Iteration for Average Cost Unichain Markov Decision Process" in *OR Computing Tools for the New Millennium*, M. Laguna and J. Velarde, editors, Kluwer Academic Publishers, 2000, pp. 161-182,
[4] W. Cooper, S. Henderson, and M. Lewis, "Convergence of Simulation-Based Policy Iteration", *Probability in the Engineering and Informational Sciences*, Vol. 17(2), 213-234, 2003.
[5] Y. C. Ho, R. S. Sreenivas, and P. Vakili, "Ordinal Optimization of DEDS," *Journal of Discrete-event Dynamic Systems*, Vol.2, No.2, pp. 61-88, 1992.
[6] V. Fabian, *Stochastic Approximation*, *Optimization Methods in Statistics*, Edited by J. S. Rustagi, *Academic Press*, New York, 1971.
[7] H. J. Kushner, and D. S. Clark, *Stochastic Approximation for Constrained and Unconstrained Systems*, Springer-Verlag, 1978.
[8] L. Dai, "Convergence Properties of Ordinal Comparison in the Simulation of Discrete-event Dynamic Systems," *Journal of Optimization Theory and Applications*, Vol. 91, No.2, pp. 363-388, November 1996.
[9] L. Dai, and C. H. Chen, "Rate of Convergence for Ordinal Comparison of Dependent Simulations in Discrete-event Dynamic Systems," *Journal of Optimization Theory and Applications*, Vol.94, No.1, July, 1997.
[10] B.-W. Hsieh, C.-H. Chen, S.-C. Chang, "Scheduling Semiconductor Wafer Fabrication by Using Ordinal Optimization-based Simulation," *IEEE Trans. on Robotics and Automation*, Oct ,2001.
[11] B.-W. Hsieh, S.-C. Chang, C.-H. Chen, "Efficient Selection of Scheduling Rule Combination by Combining Design of Experiment and Ordinal Optimization-based Simulation," *Proceedings of ICRA2003*, Taipei, Sept., 2003.
[12] S. M. Ross, *A First Course in Probability*. Englewood Cliffs, NJ:Prentice-Hall, 1984.
[13] C. H. Chen, "A Lower Bound for the Correct Subset-Selection Probability and Its Application to Discrete-event System Simulations," *IEEE Transactions on Automatic Control*, Vol. 41, No. 8, pp. 1227-1231, August, 1996.
[14] C.H. Chen, H. C. Chen, and E. Yucesan," Computing Efforts Allocation for Ordinal Optimization and Discrete Event Simulation," *IEEE Transactions on Automatic Control*, May 2000.
[15] R. A. Howard, *Dynamic programming and Markov Processes*, Technology Press of Massachusetts Institute of Technology, 1960.
[16] Averill M. Law, and W. David Kelton, *Simulation Modeling and Analysis*, McGraw-Hill, 1991.
[17] S. H. Lu, D. Ramaswamy, and P. R. Kumar, "Efficient Scheduling Policies to Reduce Mean and Variance of Cycle-Time in Semiconductor Manufacturing Plants, " *IEEE Transactions on Semiconductor Manufacturing*, Vol. 7, No. 3, August 1994.
[18] Bitran, G. R. and D. Tirupati, "Multiproduct queueing networks with deterministic routing: decomposition approach and the notion of interference," *Management Sci.*, 34 (1988), 75-100.
[19] D. Gross and C. M. Harris, "Fundamentals of Queueing Theory"
[20] G. Horvath, "Approximate Waiting Time Analysis of Priority Queues".
[21] M.-D. Hu, S.-C. Chang, "Translating Overall Production Goals into Distributed Flow Control Parameters for Semiconductor Manufacturing," *Journal of Manufacturing Systems,* Vol. 22, No. 1, 46-63, 2003.

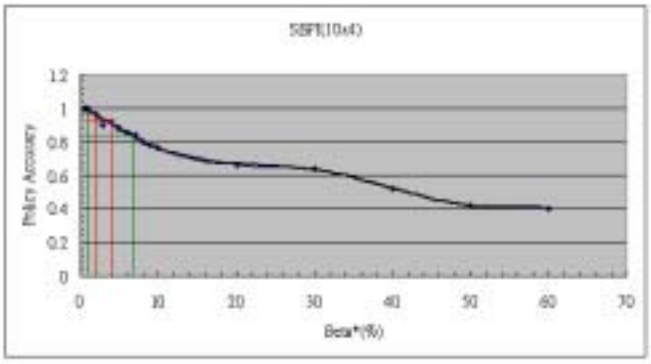

Fig 2.1 Policy accuracy vs. 1-CTG estimation accuracy 彑

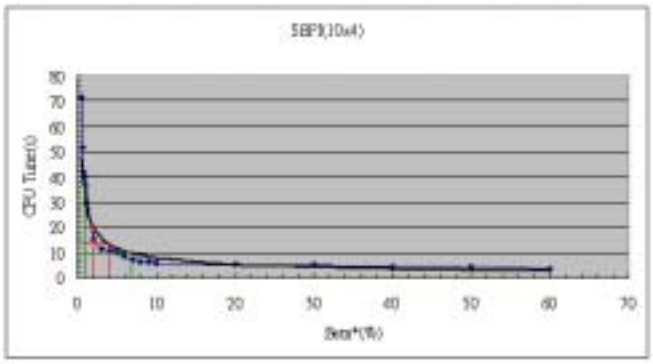

Fig 2.2 CPU time vs. Estimation accuracy β^*

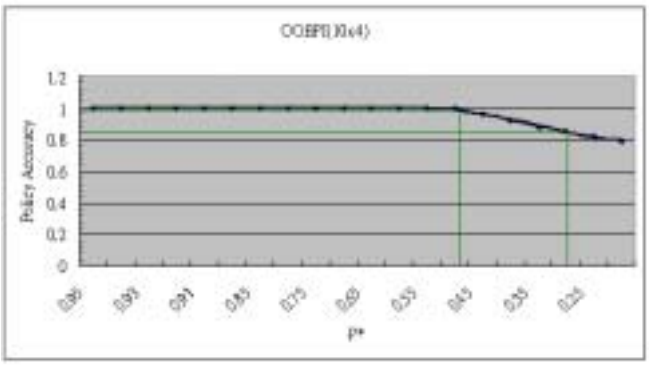

Fig 5.1 P^* vs optimal policy accuracy

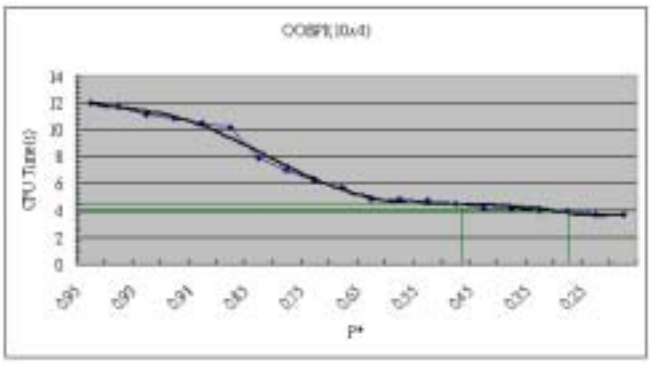

Fig 5.2 P^* vs simulation time

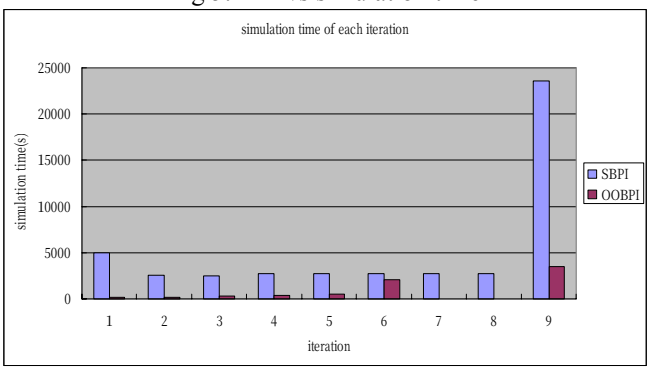

Fig 5.3 simulation time of OOBPI vs. SBPI

Enter →	1 →	2 →	3 →	8 →	10 →	
	1 →	2 →	6 →			
	1 →	2 →	3 →	8 →	9 →	
	1 →	2 →	3 →	8 →	10 →	
	1 →	6 →	11 →	1 →	2 →	5 →
	1 →	2 →	3 →	8 →	9 →	
	1 →	2 →	3 →	8 →	10 →	
	1 →	6 →	11 →	1 →	2 →	5 →
	1 →	2 →	3 →	8 →	9 →	10 → 11 →
	1 →	2 →	4 →	8 →	3 →	
	1 →	2 →	7 →	8 →	10 →	
	1 →	2 →	12 →	Exit		

Figure 6.2: Process Flow of FAB1

Product	Number of Operations	Process Routing Sequence
1	8	1, 2, 4, 2, 9, 10, 11, 13
2	9	1, 2, 5, 2, 8, 9, 10, 11, 13
3	9	1, 2, 6, 4, 2, 9, 12, 11, 13
4	9	1, 2, 7, 4, 2, 9, 10, 11, 13
5	8	1, 2, 4, 12, 2, 9, 2, 13
6	8	1, 2, 5, 12, 2, 9, 7, 13
7	8	1, 2, 6, 12, 2, 8, 2, 13
8	12	1, 2, 3, 7, 4, 12, 2, 8, 6, 9, 2, 13
9	13	1, 2, 3, 5, 4, 6, 12, 2, 8, 2, 10, 6, 13
10	13	1, 2, 3, 6, 2, 4, 12, 7, 2, 9, 11, 5, 13

Table 6.1 Product Routing Sequence of FAB2

MG	# of Machines	Service Time Distribution	MPT (hr/lot)	Utilization %*
1	1	Uniform	0.78	78.0
2	1	Uniform	0.25	62.5
3	1	Erlang Order 2	1.667	50.0
4	1	Exponential	1.057	74.0
5	1	Erlang Order 3	1.700	68.0
6	1	Erlang Order 4	0.950	57.0
7	1	Uniform	1.775	71.0
8	1	Uniform	1.875	75.0
9	1	Erlang Order 2	1.175	94.0
10	1	Erlang Order 3	1.800	72.0
11	1	Exponential	1.430	71.5
12	1	Erlang Order 4	0.750	52.5
13	1	Uniform	0.870	87.0

* Utilization % = $\sum_{i=1}^{10}\left[\frac{0.1(\#ofVisits)(MPT)}{\#ofMachines}\right]\times 100$.

Table 6.2 Machine Group Data of FAB2

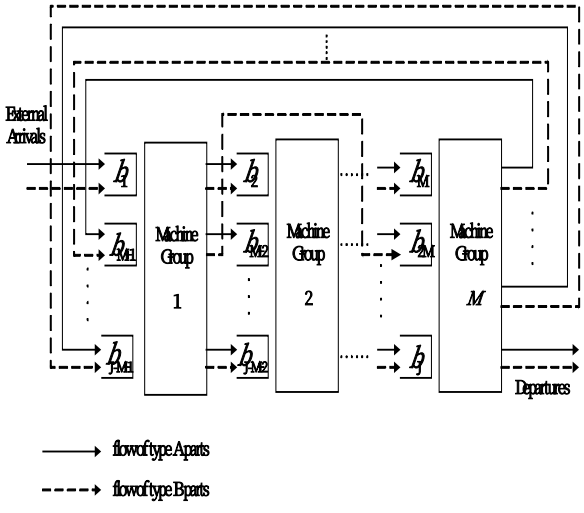

Figure 6.2: Priority Open Queueing Network (Fab Example)

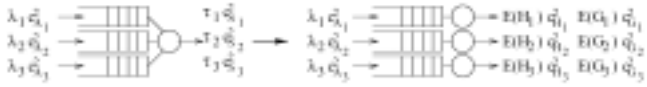

Figure 6.3 GI/G/1 Special Service Queue