MOTIVATION
Semi Conductor Wafer Fabs

A Semiconductor Wafer

Fabrication of a single oxide layer

Clean → Oxidation → PhotoLithography

Photoresist Strip → Ion Implantation or metal deposition → Etching
A Re-Entrant Flow Line

From an HP pilot plant in California, example used by P.R. Kumar (1992)
Clean Room with Processing Bays:

A Clean Room with Production Bays

Complicating factors:

Batching, setups, variability between steps.

We focus on the reentrant structure, we want to control the flow of wafers along the whole line.
Wafer fab cost: $3 \times 10^9$.
Return in 3 years
Wafer cycle time: 6 weeks
WIP $180 \times 10^6$

Modern Mega Fab ships 10,000 wafers per week
WIP 60,000 wafers.

Thousands of wafers spread out over hundreds of processing steps at dozens of machines.

Currently ad-hoc control, using simulation software.

Objective is not just to save costs but to establish better control and predictability, improve overall performance, and increase sales and revenues.

Figure 1. Generic 130-nm Semiconductor Process Flow Showing Massively Reentrant Product Flows
The Job Shop Scheduling Problem

Machines \( i = 1, \ldots, I \),
Routes \( r = 1, \ldots, R \), Steps \( K_r \).

Jobs \( j = 1, \ldots, J \)
  Release time: \( A(j) \)
  Route: \( r = r(j) \)
  Processing times \( X_{r,1}(j), \ldots, X_{r,K_r}(j) \)
  On machines \( \sigma(r,1), \ldots, \sigma(r,K_r) \)

Schedule \( s_{r,k}(j), t_{r,k}(j) \) start and finish time for all ops.

Objective: Minimize Makespan (last job completion)
           Minimize Weighted Flowtime (cycle time)

Job Shop Combinatorial Optimization Approach:
  NP-hard and conceptually problematic non-robust.
Multi-Class Queueing Networks:

Nodes (machines): \( i = 1, \ldots, I \)
Classes, queues, buffers, steps: \( k = 1, \ldots, K \)
Constituencies: \( i = \sigma(k), \quad k \in C_i \)

Jobs \( j = 1, \ldots, N \)
Arrivals: \( A(j), \) renewal process, rate \( \alpha \)
Processing: \( X_k(j), \) i.i.d. mean \( m_k \)
Routing: \( k \rightarrow l, \) Bernoulli, \( P_{k,l} \)

Queue balance dynamics equations
\[
Q(t) = Q(0) + A(t) - D(t)
= Q(0) + A(t) + (P_{\mathbf{S}(T(t))} \odot I) \mathbf{S}(T(t)) \geq 0
\]

Objective: Optimal expected steady state cost rates.

Multiclass Queueing Network, MDP Approach:
Intractable and conceptually problematic no steady-state.
On-Line Finite Horizon Scheduling

INSTEAD:
finite horizon (job shop scheduling), on-line using $Q(t)$ (queueing).

FLUID APPROXIMATION
Obtained from the scheduling problem by
- Relaxing integrality of buffer levels
- Relaxing integrality of machines
- Relaxing integrality of jobs and operations

Obtained from MCQN by scaling initial fluid, time, and state
$$q(t) = \lim_{n} \frac{1}{n} Q(nt), \quad n = \|Q(0)\| \to \infty$$
$$q(t) = q(0) + \alpha t - (I - \hat{P} \hat{C} \mu \cdot T(t))$$

Step 1: Formulate a fluid problem. The fluid solution provides an approximate lower bound on the optimal cost.

Step 2: Solve fluid problem as an SCLP. Solution is highly informative.

Step 3: Use a fluid following heuristic. Difference between heuristic and fluid bounded in probability.

MANY JOBS $N$ IN A FIXED SYSTEM --- WE EXPECT:

WIP costs
$$V^{Fluid} \approx V^{Optimum} \leq V^{Heuristic} = O(N^2)$$
$$P(V^{H} - V^{F} > O(\sqrt{N})) \xrightarrow{N \to \infty} 0$$

Makespan
$$M^{Fluid} \leq M^{Optimum} \leq M^{Heuristic} = O(N)$$
$$\text{in general: } P(M^{H} - M^{F} > O(\log N)) \xrightarrow{N \to \infty} 0$$
$$\text{some cases: } P(M^{H} - M^{F} > O(1)) \xrightarrow{N \to \infty} 0$$
Fluid Formulation -- Example

Queue balance dynamics equations

\[
\begin{align*}
Q_1(t) &= Q_1(0) + A_1(t) - S_1(T_1(t)) \\
Q_2(t) &= Q_2(0) + 0 + -S_2(T_2(t)) + S_1(T_1(t)) \geq 0 \\
Q_3(t) &= Q_3(0) + 0 + -S_3(T_3(t)) + S_2(T_2(t))
\end{align*}
\]

Where the total processing times devoted to the buffers satisfy

\[
T_k(0) = 0, \quad T_k(t) \uparrow, \quad T_1(t) + T_3(t) \leq t \\
T_2(t) \leq t
\]

The fluid balance dynamics equations

\[
\begin{align*}
q_1(t) &= q_1(0) + \alpha t - T_1(t)/m_1 \\
q_2(t) &= q_2(0) + 0 + -T_2(t)/m_2 + T_1(t)/m_1 = \\
q_3(t) &= q_3(0) + 0 - T_3(t)/m_3 + T_2(t)/m_2
\end{align*}
\]

\[
\begin{align*}
q_1(0) &= q_2(0) + 0 + \int_0^t -u_2(s) + u_1(s) \, ds \\
q_3(0) &= 0 - u_3(s) + u_2(s)
\end{align*}
\]

Where the total processing times devoted to the buffers satisfy

\[
\begin{align*}
u_k(t) &\geq 0, \quad m_1u_1(t) + m_3u_3(t) \leq 1 \\
m_2u_2(t) &\leq 1
\end{align*}
\]
The fluid problem is of the form:

\[
\text{max } \int_0^T (T-t)c\varphi(t)\,dt \\
\int_0^t G\varphi(s)\,ds + x(t) = a + \alpha t \\
H\varphi(t) = b \\
x, u \geq 0, \quad 0 < t < T
\]

where for the previous example:

\[
G = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \quad a = \begin{bmatrix} Q_1(0) \\ Q_2(0) \\ Q_3(0) \end{bmatrix}, \quad \alpha = \begin{bmatrix} \alpha \\ 0 \end{bmatrix},
\]

\[
H = \begin{bmatrix} m_1 & 0 & m_3 \\ 0 & m_2 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix},
\]

\[
c\varphi = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.
\]

Integration by parts:

\[
\int_0^T (x_1(t) + x_2(t) + x_3(t))\,dt = T(a_1 + a_2 + a_3) - \int_0^T (T-t)u_3(t)\,dt
\]

Hence:

\[
\text{max } \int_0^T (x_1(t) + x_2(t) + x_3(t))\,dt
\]

is equivalent to

\[
\text{min } \int_0^T (T-t)u_3(t)\,dt
\]
Fluid Solution -- Example

For the 2 machine 3 buffer example, $m_2 > m_1 + m_3$:

(Solution of this example is joint with Florin Foram, 1992)
## Example -- 5 machine 20 buffer re-entrant

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Let the fluid solution be given by: \[ q_k(t) \]

Let the actual number of jobs in the system be \( Q_k(t) \)

Define:

\[ q_k^\oplus(t) \] total fluid to leave buffer \( k \) by time \( t \), in the fluid solution (fluid departures)

\[ Q_k^\oplus(t) \] total number of jobs that have left buffer \( k \) by time \( t \), (actual departures)

The quantity \( q_k^\oplus(t) - Q_k^\oplus(t) \) represents how much our actual schedule lags behind the fluid solution.

Priority measure

\[ \rho_k(t) = \frac{q_k^\oplus(t) - Q_k^\oplus(t)}{q_k^\oplus(t)} \]

**Greedy Fluid Algorithm (GFA):** Machine \( i \) works on buffer:

\[ k_i^* = \arg \max \{ \rho_k(t) : k \in C_i, Q_k(t) > 0 \} \]
**Does Fluid Heuristic work:**

**Supporting Evidence**

Makespan in Job shop has simple fluid solution.

We can approximate it extremely well:

Papers of Bersimas and Gamarnik: $O(\sqrt{N})$

Paper of Dai and W: $O(\log N)$

Experimental evidence: $O(1)$

Single bottleneck as well as multiple bottleneck

MCQN with infinite buffers

Can be extended to weighted flowtime objective?
Minimizing Fluid Makespan.

Calculate for each machine the total amount of work that it needs to do, before the whole system can be empty:

\[ T_i \quad i = 1, \ldots, I \]

Take maximum over all the machines:

\[ T^* = \max_i T_i \]

Consider any of the buffers in the system, which starts with level \( q_k(0) \)

Optimal (not unique) Makespan solution is:

\[ q_k(t) = q_k(0) \left( 1 - \frac{t}{T^*} \right) \]

The fluid solution provides a lower bound for the discrete schedule.
Cyclic Scheduling with Safety Stocks

To keep Bottleneck machine busy throughout full cycles: Build up SAFETY STOCKS, $S$ in each buffer.

Then perform Cycles, initiated by the Bottleneck Machine.

Bottleneck idles if maximal queue length exceeds $S$.

**Theorem** (Dai & W): $X_{r,o}(j)$ i.i.d. with means $m_{r,o}$ and

1. $i^*$ is a unique, single, bottleneck.
2. $X_{r,o}(j)$ possess exponential moments.

Let $T^*, T^H$ be the (random) lower bound and heuristic makespans.

There exist $C_1, C_2$ such that for all $N$:

With safety stocks $S = C_1 \log N$,

$$P(T^H - T^* > C_2 \log N) < 1/N$$
Job Shop Simulation Study
(Yoni Nazarathy)

http://rstat.haifa.ac.il/~yonin/thesis/jobshopsim/shopsim.html
Assume the first buffer has an unlimited supply of jobs.

Consider the system under LBFS policy: Buffer 1 is served only if all the other buffers of machine $\sigma_1$ are empty.

The fluid picture now is:

- Queue in buffer 1 is $\infty$
- Buffers $a^{(L-1)},...,a^{(1)}$: $\rho > 1$
- All other queues are stable
- No starvation after $T_{\text{starve}}$
- Stable time for $T_{\text{runout}}$
Multiple Bottlenecks:

We found that high volume Job Shops with more than one bottleneck may also be scheduled with a gap of $O(1)$.

We also found that high volume job shop problems with random routes, where each route has $K$ processing steps, and the processing times of all steps are i.i.d. (i.e. all machines are bottlenecks), can also be scheduled with a gap of $O(1)$.

How is that possible?

Consider MCQN with Infinite Virtual Buffers.
A Push Pull System
(Anat Kopzon):
Consider the following Two Machine Two Routes System:

Top Route goes from machine 1 to machine 2.
Bottom Route goes from machine 2 to machine 1.
Machine \(i\) works *feeds* machine \(3-i\) at rate \(\lambda_i\).
Machine \(i\) *serves* its own queue at rate \(\mu_i\).

**Flow balance:**
To keep machines busy all the time and be stable:

\[ \alpha_i \text{ proportion of time machine } i \text{ is feeding, } 1 - \alpha_i \text{ it is serving.} \]

The flow rates satisfy:

\[ \nu_1 = \alpha_1 \lambda_1 = (1 - \alpha_2) \mu_2 \]
\[ \nu_2 = \alpha_2 \lambda_2 = (1 - \alpha_1) \mu_1 \]

Hence:

\[ \nu_1 = \frac{\lambda_1 \mu_2 (\mu_1 - \lambda_2)}{\mu_1 \mu_2 - \lambda_1 \lambda_2} \]
\[ \nu_2 = \frac{\lambda_2 \mu_1 (\mu_2 - \lambda_1)}{\mu_1 \mu_2 - \lambda_1 \lambda_2} \]

This can be achieved by a stable policy!
Lecture outlines:

Lecture 2: Scheduling and Queueing

talk about some simple scheduling problems, including calculation of performance measures and probabilistic analysis, and discussion of some NP hard problems.

Single server queue - mainly first moments, priorities, fluid and diffusion approximations.

Lecture 3: Queueing Networks

single class queueing networks, dynamics, fluid and diffusion approximations.

multiclass networks, stability via fluid models, MDP formulation and fluid solutions.

Lecture 4: Fluid solutions

stability and instability of some MCQN under various policies. SCLP formulation. shortest time problems. minimum flowtime examples.

Lecture 5: Solution of Separated Continuous Linear Programs

linear programming background, the dual of SCLP, weak duality and complementary slackness, associated LP, feasibility and unboundedness, non-degeneracy, theorem on the structure of the optimal solution, a simplex type algorithm to solve SCLP.

Lecture 6: Fluid imitation heuristics

the job-shop makespan problem, solutions for high volume job shops - cyclic solutions, safety stocks, arbitrary product mix. single bottleneck - LBFS for the reentrant line, other high volume job shops. MCQN with virtual infinite buffers.

Lectures 7,8: To be decided within class
How will this be used in the fab

Current control of fabs:
Local optimization,
JIT, conwip, CANBAN,
MRP
Simulation.

We can help:
The fluid solution can be used to set goals for single stations, which can than try to reach these goals in an optimal way. This sort of hierarchical control has many advantages

Our approach highlights the difference between the flowshop setup and the much more complicated reentrant line of job-shop setup. In particular, MCQN exhibit a whole range of phenomena unknow in tandem queues. Hence, the JIT etc. approaches which worked for production lines in the car industry probably fall short in the wafer fab.

MRP does not take plant state into account. We can get MRP type output for batches from our fluid solution.

Simulation: What we get as a fluid solution can replace a detailed simulation for many purposes. We can use the fluid model to do sensitivity analysis, answer what if questions, get forecasts, and do planning and design.