Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

SYMBIOTIC SIMULATION FOR OPTIMISATION OF TOOL OPERATIONS IN SEMICONDUCTOR MANUFACTURING

Heiko Aydt

School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798

Wentong Cai

School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798

Stephen J. Turner

School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798

Boon Ping Gan

D-SIMLAB Technologies Pte Ltd
8 Jurong Town Hall Road #30-04 JTC Summit
Singapore 609434

ABSTRACT

A symbiotic simulation-based problem solver agent is proposed that can be used to automatically solve decision making problems regarding the operations of the various tools of an entire semiconductor manufacturing plant (fab). In comparison with common practice decision making, performed by human operators, the advantage of the symbiotic simulation-based approach is its ability to simulate how decisions will affect operations in different parts of the fab. Previous work has been concerned with the optimization of a single tool group. Here, we show that our approach can also be applied to control an entire fab which typically involves several hundred tools. Unlike other approaches, ours is not limited to a set of pre-defined decision making policies. Instead, the problem solver agent can directly schedule setup changes for an arbitrary number of tools. Experiments show that higher throughput can be achieved by using our approach as compared to common practice decision making.

1 INTRODUCTION

Semiconductor manufacturing is a highly complex and asset intensive process. In order to stay competitive, manufacturers are required to continuously improve their manufacturing process and invest several million US dollars for automation solutions (Potoradi et al. 2002, Gan et al. 2006). An important issue in this context is the efficient utilization of resources. The tools used in semiconductor manufacturing can cost up to US$ 2 million (Scholl and Domaschke 2000). Highly efficient utilization of tools is therefore crucial. This can be achieved by improving decision making concerned with the operation of the tools. In our previous work (Aydt et al. 2008) we have already highlighted the need for automation in semiconductor manufacturing and described an application, based on symbiotic simulation (Fujimoto et al. 2002), which improves the performance of a group of wet benches (a particular kind of tool used to clean wafers from residues after certain fabrication steps). In contrast, the work presented in this paper is concerned with the optimization of an entire semiconductor manufacturing plant (fab). As we will explain later, a fab consists of several hundred tools (as compared to 8 in our previous work) which makes the problem several magnitudes more complex.

Symbiotic simulation is a paradigm in which a physical system is closely coupled with a simulation system by means of sensors and actuators (Fujimoto et al. 2002). The simulation system benefits from real-time sensor data which can be used to initialize and drive high-fidelity simulations of the physical

978-1-4577-2109-0/11/$26.00 ©2011 IEEE

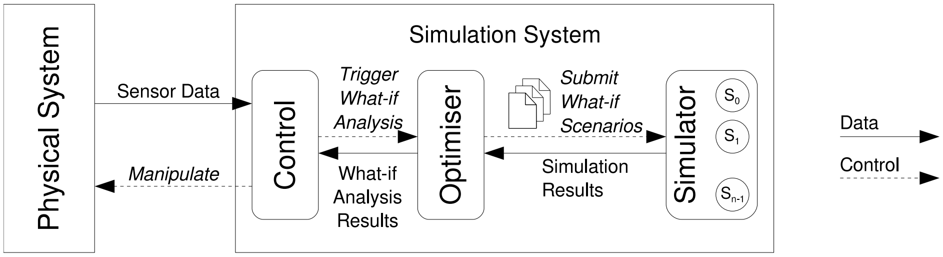


Figure 1: Overview of a symbiotic simulation system for decision support and control. The simulation system is coupled with the physical system. The optimizer is responsible for creating candidate solutions which are evaluated by means of simulations.

system. The physical system, on the other hand, may benefit from the ability of the simulation system to quickly evaluate alternative decision options. Figure 1 illustrates a symbiotic simulation system. The process of evaluating a number of alternative what-if scenarios (each representing an alternative decision options) is referred to as the what-if analysis process. However, unless the problem is trivial, there is not enough time to evaluate all possible what-if scenarios, i.e., an exhaustive search for the optimal decision option is infeasible. For this reason it is necessary to use an appropriate optimization method that is capable of identifying solutions without having to evaluate all what-if scenarios. Since symbiotic simulation is used for short-term decision making there is only a limited amount of time available to solve a problem. In addition, finding a reasonably good solution just in time is often more important in the context of real-world problems than finding an optimal solution. These issues have to be taken into consideration when selecting an appropriate optimization method. As we will explain later in Section 3, we have used an evolutionary computing approach.

Scheduling of wafer lots to production resources (e.g., tools) is a crucial performance issue in semiconductor manufacturing. Various work has been concerned with this issue. For example, a large body of work is concerned with the evaluation of dispatching policies. These policies decide, according to some criteria, in which sequence wafer lots are being assigned to production resources. Research in this area often focuses on a single type of tool, such as stepper tools (Wu et al. 2006) or furnaces (Akçali et al. 2000), for instance. The work by Zhang et al. (Zhang, Jiang, and Guo 2008) is notable as it employs the same concepts that are essential to symbiotic simulation. They describe a system in which a scheduler is monitoring and controlling a fab. The scheduler determines when a new rule combination (dispatching rule + rework strategy) has to be used. Rule optimization is done with the support of a simulator that is employed by the scheduler to evaluate the performance of a given combination of dispatching rule and rework strategy. Once the best combination has been identified, it is implemented in the fab. Essentially, what Zhang et al. describe is a symbiotic simulation decision support and control system.

Dispatching (i.e., assignment of a wafer lot to a tool) is dependent on the current setup of a tool. A wafer lot can only be dispatched if the setup of a particular tool is compatible with the one needed by the wafer lot. Changing tool setups can be time consuming. Frequent re-configurations should thus be avoided in order to reduce inefficient tool utilization. This can be achieved by employing dispatching policies that explicitly consider setups (Sethi, Chu, and Yan 1999). However, while policies can be used to locally optimize the performance of parts of the fab, they are not well suited to optimize the entire fab. A fab is a highly complex system and changes to one part of the system is likely to affect other parts. In order to evaluate how changes to one part of the system will affect other parts, simulations are required.

Decision making regarding tool configuration is closely related to scheduling as wafers can only be processed if the required tool is configured to do so. Depending on the product, different wafers may need different setups. In this paper, we apply our symbiotic simulation-based approach in order to improve fab throughput by scheduling the change of setups more efficiently. The advantage of our approach is the ability to predict the effects of setup changes on the overall performance of the fab. Existing work, such as the one by Zhang et al., often aims at changing setups of tools only as a result of dispatching wafer lots to production resources that require a different setup. In contrast, our approach directly decides on the schedule for changing setups, i.e., it does not rely on alternating between different scheduling policies. Instead, the problem solver agent has the freedom to change the setup of any tool at any point of time. In this paper, we show that our approach improves the performance of the physical system in terms of fab throughput when comparing to common practice decision making.

This paper is structured as follows. In Section 2 we discuss the model of the semiconductor manufacturing system. In Section 3 we discuss the design of the problem solver agent used for this application. In addition, we discuss how the what-if analysis process can be specialized, depending on the current operating conditions. In Section 4 we discuss a number of experiments with focus on the impact of using a problem solver agent to complement local decision making and the impact of specialization on the performance of the what-if analysis process. We conclude this paper by discussing the application example and summarize the contribution of this paper in Section 5 and Section 6, respectively.

2 SEMICONDUCTOR MANUFACTURING MODEL

The model of the semiconductor manufacturing system considered in this paper is based on information about a real-world fab and has been developed with the support from domain experts. However, the model does not exactly reflect any particular real fab for proprietary reasons. Some details have not been available for developing the model. For example, this includes the exact number of tools as well as precise timings of the various processing steps. The purpose of this application is to demonstrate the applicability of our approach to decision support in semiconductor manufacturing. In particular, the advantage of symbiotic simulation-based problem solving in comparison with common practice decision making is being investigated. For this purpose, it is important to have a model which realistically reflects the complexity of the semiconductor manufacturing process. It does not necessarily have to reflect an existing fab exactly with all its proprietary details. Therefore, for creating the model, we have focused on a realistic degree of complexity rather than the exact representation of one particular real-world fab.

2.1 Manufacturing Process

A fab can produce multiple products concurrently. From the perspective of the manufacturing process, products are characterized by the different processing steps that are needed to turn the raw wafer into the finished product. A total of 18 different products is considered in this application. Each product is associated with a workflow that represents a sequence of processing steps. The workflow indicates the exact order in which wafers have to be processed, the tool and setup that have to be used for each processing step, as well as the exact processing times needed to complete a step. The total number of processing steps required by various products is ranging between 169 and 347. On average, a product has to go through 252 different steps during its manufacturing process. In this application, we assume that all products are processed concurrently, i.e., wafers from different workflows are competing for the same production resources.

The time required for a wafer to complete the entire manufacturing process is commonly referred to as cycle time. We distinguish between the theoretical cycle time, which is the total processing time of all steps without any delays, and the actual cycle time, which also includes delays caused by transportation within the fab, for example, or unavailability of processing resources. The average theoretical cycle time is 9.8 days. The shortest and longest theoretical cycle time is 3.4 days and 14.8 days, respectively. An overview of the various products is illustrated in Table 1.

Depending on the processing step, wafers are processed either one by one or by the lot. Some tools also allow batch processing, i.e., to process multiple wafer lots concurrently. Although wafers might be processed one by one, they are always transported from one tool to another in their cassettes. Loading and unloading of wafers from their cassette is done fully automatically by the corresponding robotic handlers in the tool. This reduces the need of human intervention which would otherwise lead to impaired quality

Table 1: Overview of the various products, including the number of processing steps and the theoretical cycle time for each product.

Product Id	# of Steps	Theoretical Cycle Time	Product Id	# of Steps	Theoretical Cycle Time
P0001	169	7.6 days	P0010	239	8.9 days
P0002	240	8.8 days	P0011	239	8.9 days
P0003	347	14.8 days	P0012	334	14.8 days
P0004	347	14.8 days	P0013	233	9.0 days
P0005	345	14.8 days	P0014	196	7.4 days
P0006	334	14.8 days	P0015	196	7.4 days
P0007	236	10.1 days	P0016	233	9.0 days
P0008	345	14.8 days	P0017	206	7.8 days
P0009	240	8.8 days	P0018	225	3.4 days

due to improper handling. In the simulation model, we do not explicitly represent wafers. Instead, the smallest entity are wafer lots and for the remainder of this chapter, we will use wafer lots (with exactly 25 wafers) as the unit for denoting workload and throughput.

Various kinds of tools are used in the semiconductor manufacturing process. This includes tools such as furnaces, lithography tools, cluster tools, wet benches, and a number of other tools. The fab model has a total number of 286 stations, each of which is equipped with a single tool. They are organized into 65 station families. Each station family has a storage capacity for temporarily keeping wafer lots until a tool becomes available for processing waiting lots. Depending on the tool, processing is done either by lots or batches of lots. In case a tool is capable of processing wafer lots in batches, corresponding batching policies are used that determine how batches are composed. An overview of the various tool types used in this application is illustrated in Table 2.

Tools can be operated with different setups. However, although a tool may technically be capable of operating under a certain setup, it may not be certified for this purpose. Certification is an important measure to maintain the required quality of wafers. The certification procedure can last several weeks and is thus not subject to operational decision making which is usually in terms of hours and days. Therefore, from this application's perspective, certifications cannot be changed. The number of setups supported by a certain tool varies greatly. While some tools are certified to support up to 27 different setups, other tools support only a single setup. The average number of certified setups per tool is 6. Changing the setup of a tool often requires downtime (i.e., setup time) during which the tool is not available for processing any wafers.

Not all tools require setup times. For example, in case of wet benches, a "setup" refers to a particular recipe according to which wafers are treated. Wet benches are tools that are used to clean the wafer after certain fabrication steps to remove residues which would otherwise affect the quality of the wafer. This cleaning process requires wafers to be processed in a number of chemical liquids for a precisely specified time period. A recipe is thus a unique sequence of chemicals and specific processing times. Different wafers can be processed according to different recipes (i.e., different chemical sequences and timings). However, a wet bench is equipped with a fixed set of chemical baths. Although the liquids have to be replaced from time to time (maintenance), they are not changed (i.e., replaced by a different chemical). Therefore, there is no downtime (i.e., setup time) required when processing wafers according to a different recipe.

In contrast, furnace tools may operate at different temperatures (in this case "setup" refers to a temperature setting). Wafers have to be processed at a certain temperature for a certain period of time. To assure quality, it is important that the temperature is not fluctuating. Therefore, if wafers require different processing temperatures (i.e., different setups), a furnace needs to change its operating temperature. The

Table 2: Overview of the various tool types, number of station families, total number of tools in the fab model, the maximum batch size in terms of wafer lots (if the tool supports batching), and the setup time in minutes (if applicable to that tool). Actual setup times are assumed to follow a normal distribution. Therefore, corresponding mean setup times and standard deviations are given.

Tool Type	# of Families	# of Tools	Batching	Setup Time
				Mean (Std.Dev.)
Backside Clean	1	2	N/A	15 (5)
Cluster tool	23	108	N/A	15 (5)
CMP tool	3	5	N/A	N/A
DNS Clean	3	3	N/A	15 (5)
Lithography	1	5	N/A	20 (3)
Furnace	10	33	6 lots	35 (8)
Hardbake	1	2	6 lots	15 (5)
Hotplate	1	2	6 lots	15 (5)
I-Line	1	15	N/A	20 (3)
Implanter	3	21	N/A	N/A
Laser Marker	1	2	N/A	N/A
Metrology	6	52	N/A	N/A
Wet bench	11	36	2 lots	N/A

heating up/cooling down process takes some time (i.e., the setup time) during which no wafers can be processed. An overview of the various setup times of all tool types (if applicable) is illustrated in Table 2.

Despite efforts towards automated operations, operational decisions regarding configuration of tools is made by human operators. The model reflects this fact: each station family is assigned an operator who is responsible for changing the setups of the various supervised tools. Figure 2 illustrates an example of a station family with three tools, each of which is equipped for different setups.

2.2 Common Practice Decision Making

Common practice decision making is distributed (i.e., operators make decisions regarding the tools under their supervision) and involves local optimization (i.e., each operator tries to optimize the performance locally in terms of throughput, for example). Although operators of different station families can communicate with each other and coordinate their actions, they cannot fully anticipate how local decision making is going

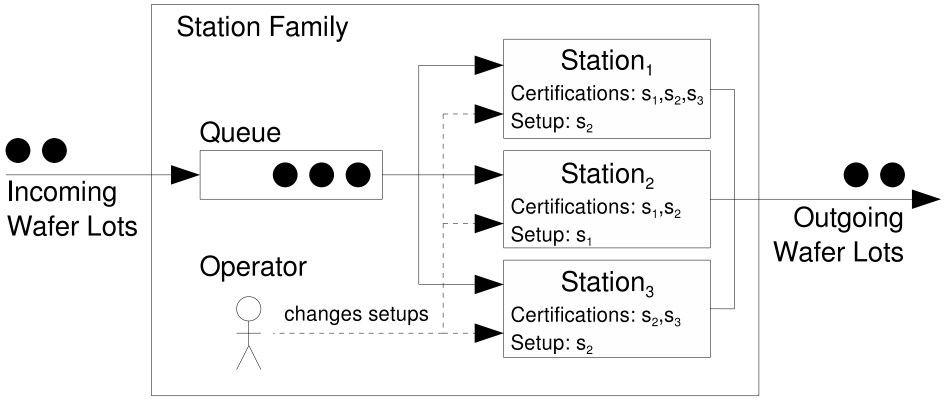


Figure 2: Example of a station family with three tools, each of which is certified for different setups.

to affect different parts of the fab. This is due to the complexity of the manufacturing process. Simulations are required to see how local decision making affects other parts of the fab. Therefore, centralized decision making, using full knowledge about the state of the fab, is beyond the capabilities of human operators.

The common practice decision making approach, used in our model, is based on information provided by the industry and meant to emulate decision making by human operators in a fab. Operators generally distinguish between priority and non-priority tools. Priority tools are the ones that are more critical to the manufacturing process. This includes cluster tools, lithography tools, furnaces, implanters, and wet benches. Operators watch priority tools more closely and act faster upon changing operation conditions as compared to non-priority tools. The setup of non-priority tools is changed only once every working shift (i.e., once every 8 hours). In contrast, priority tools may be re-configured every hour if the situation requires. Operators analyze the setup demand of the waiting lots in the queues of the station families supervised by them. For our model we assume that the goal of the operator is to match setup demand and supply of a station family f by re-configuring as many tools as necessary in order to minimize the discrepancy $\delta_{DS}(f)$ between the setup demand and supply.

We assume that operators are free in their choice of selecting tools within a station family for re-configuration. Therefore, every time an operator checks the current operating situation of a station family, multiple tools may be selected for re-configuration in order to match demand and supply. Human operators often apply a "fire fighting" approach. We model operator behavior as follows:

1. Determine the critical setup for each station family, i.e., the setup which is highest in demand compared to its supply.
2. Identify all tools within the station family that are certified for this setup but currently use a different setup.
3. Select one of the identified tools and consider the resulting demand/supply discrepancy. If changing the setup would result in an improvement, then apply the changes.

This process is repeated until no further improvements in terms of demand/supply discrepancy are possible. In the remainder of this section, we explain more formally how setup demand and supply is determined in our model.

A simple way of calculating the demand of a setup would be to count the number of lots in the queue that require this particular setup for their next processing step. However, this measure does not take into account the processing time required by this step. The processing time depends on the processing step and varies among different products and their specific process flows. The lot count does not accurately reflect the demand situation as many lots with short processing time would outweigh few lots with long processing times. Demand in terms of number of lots is thus misleading. Hence, we consider demand in terms of the waiting processing time. The waiting processing time $wpt_{f,s}$ for a particular setup s is the accumulated processing time of all lots in the queue of a station family f that require this setup. The total waiting processing time $WPT(f)$ is thus the accumulated waiting processing time for all setups that are provided by the various tools of station family f. In order to match demand and supply, we further consider the demand ratio $DR(f,s)$ and the supply ratio $SR(f,s)$ which represent the setup demand/supply relative to the total demand/supply, respectively. The demand ratio $DR(f,s)$ is calculated as follows:

$$DR(f,s) = \frac{wpt_{f,s}}{WPT(f)} \tag{1}$$

While setup demand is measured in waiting processing time, setup supply is measured in number of tools $nt_{f,s}$ in a station family f that are configured for setup s. Based on the total number of tools $NT(f)$, the supply ratio $SR(f,s)$ can be calculated as follows:

$$SR(f,s) = \frac{nt_{f,s}}{NT(f)} \tag{2}$$

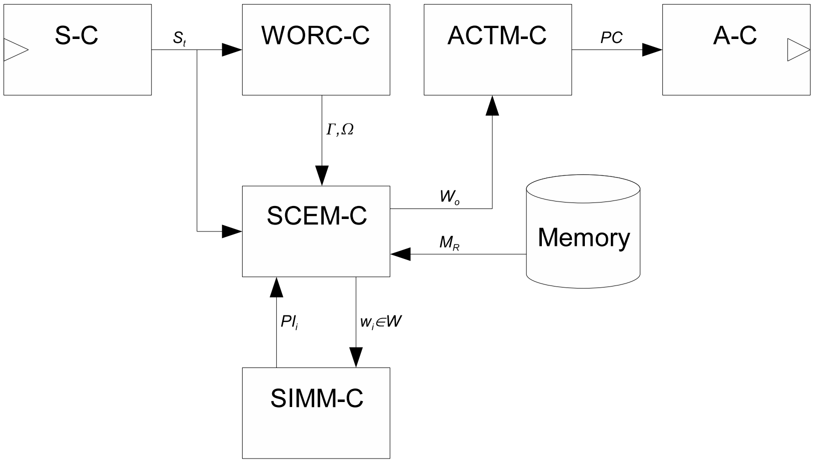


Figure 3: Design of the problem solver agent, based on a symbiotic simulation system, for control of semiconductor manufacturing equipment.

In order to match demand and supply, the objective of the operator is to minimize the discrepancy between demand and supply, which is defined as:

$$\delta_{DS}(f) = \sum_{s \in C(f)} |DR(f,s) - SR(f,s)| \tag{3}$$

where $C(f)$ is the set of all setups that are provided by the various tools of station family f.

3 PROBLEM SOLVER AGENT

The disadvantage of the common practice decision approach, described in the previous section, is the lack of complete knowledge and the ability to anticipate how decisions regarding one part of the system affect other parts. Without the use of simulation, operations are limited to local optimization based on local knowledge about the situation within a single station family. In case of human operators, these limitations are not necessarily that severe. For example, human operators have knowledge about the system and are thus capable of anticipating, to a certain extent, the likely outcome of their actions. This is particularly true for operators who can make use of experience with similar situations and problems in the past. In addition, operators can communicate with each other and coordinate their actions. Nevertheless, due to the complexity of the system, it can be assumed that operators have always only partial knowledge about the system. In contrast, a symbiotic simulation-based problem solver agent has the advantage of being able to exploit a detailed model of the physical system to make fully informed decisions using complete knowledge about the state of the physical system.

The purpose of the problem solver agent is not to completely replace distributed, local decision making. Instead, the problem solver agent overrides local decisions, whenever necessary, to improve the efficiency of the manufacturing process. Therefore, even when using the symbiotic simulation-based approach, local decisions are made according to the common practice approach explained above. In this application, we consider the case where the problem solver agent is concerned with continuously improving the overall performance of the manufacturing process. Figure 3 illustrates the design of the problem solver agent for the semiconductor manufacturing application.

The sensor component (S-C) and the actuator component (A-C) represent the link between the problem solver agent and the physical system. In practice, a fab is equipped with an information management systems that keeps track of the whereabouts and progress of the various wafer lots in the fab. This information is reflected by the state S_t of the physical system at time t. The workflow controller component (WORC-C) observes the state of the physical system and is responsible for triggering the what-if analysis. In this application, what-if analyses are performed pro-actively every 12 hours and concerned with a time period that covers the next 24 hours. The nature of the problem that is addressed by the problem solver agent may differ for each what-if analysis. Therefore, it is necessary to analyze the current operating situation and to specify the current problem in terms of 1) problem-specific information Γ and 2) objectives Ω. Since our application is concerned with setup scheduling, Γ represents information about station families that are causing problems (i.e., station families that have a high waiting processing time). In addition, Ω represents the objectives for the problem solver with respect to the current problem (e.g., minimize waiting processing time of a certain station family). For example, one what-if analysis might be concerned with solving a problem concerned with a cluster tools while another what-if analysis is concerned with improving the performance of lithography tools. In addition, the focus of the problem solver agent may be multiple tools at the same time, i.e., a what-if analysis can be concerned with more than one tool.

The information provided by Γ and Ω is exploited by the scenario management component (SCEM-C) which is responsible for creating what-if scenarios (each representing an alternative schedule for changing setups). In addition, an appropriate model of the physical system is required to create a what-if scenario. Here, we assume that a reference model M_R is always available. In practice, it might be necessary to re-calibrate or re-validate a model during runtime to adequately reflect the physical system. The purpose of the SCEM-C is to generate promising what-if scenarios while ignoring less promising ones. It is therefore important that the optimization engine (which is at the core of the SCEM-C) makes use of Γ to create only scenarios that suit the current problem. We use an evolutionary algorithm as optimization engine due to their various advantages in solving real-world problems (Fogel 2000, Bonissone et al. 2006). In particular, we make use of the evolutionary computing modeling language (ECML) (Aydt et al. 2011) to dynamically incorporate problem-specific information (represented by Γ) into the problem solving process.

More specifically, Γ represents information about which stations are performing sub-optimal and require optimization. This information is used by the evolutionary algorithm to construct genotypes that encode the schedules for the various problematic stations mentioned indicated by Γ. In our application, a genotype encodes the schedule for setup changes with a resolution of one hour. For example, consider the following genotype: $(- - - - 3 - - - - - - 1 - - - - - - - - - 2 - - -)$. Each element $(i = 1, 2, 3, \ldots, 23)$ of this genotype indicates whether a setup change is scheduled or not at time $t+i$, where t is the time when the what-if analysis has been triggered. A setup change is indicated by the setup id while a dash indicates that the setup is not changed. In this example, a total of three setup changes are scheduled 5 hours, 11 hours, and 20 hours from the time when the what-if analysis has been triggered. Note that we take into account the time needed by the problem solver agent to find a solution. Here, we assume that the what-if analysis process takes at most one hour. Therefore, the earliest time a setup change can be scheduled is at $t+1$ hours. In this simple example we considered the schedule for a single station only. Depending on the operating conditions of the physical system, the problem solver agent may have to optimize the schedule of many stations at the same time.

The framework used to realize the problem solver agent as well as the optimization engine are designed to be generally applicable (Aydt 2011), i.e., Γ and Ω are generally not limited to information regarding stations and setups or minimization of waiting processing time. Instead, Γ and Ω may represent a large variety of problem-specific information and objectives as required by the corresponding application. In our application, problem-specific information Γ is concerned with information that allows the problem solver to create solutions that focus on problematic stations. Similarly, the objectives Ω in our application are concerned with the minimization of the waiting processing time of problematic station families as well as the maximization of the overall throughput of the fab.

The simulation management component (SIMM-C) is responsible for evaluating what-if scenarios by means of simulation. For this purpose, it has to initialize simulations with the current state of the physical system (provided by S-C). For each what-if scenario w_i a corresponding performance indicator PI_i is created. The optimization engine can use the PI values of the various what-if scenarios and interpret them with respect to objectives Ω in order to decide which scenarios can be discarded. The remaining what-if scenarios are processed (by standard evolutionary algorithm variation operators) in order to obtain another set of what-if scenarios which can be evaluated by the SIMM-C. This iterative optimization process is performed until a given computing budget has been exhausted. The set W_o of optimal what-if scenarios is forwarded to the action management component (ACTM-C) which is responsible for selecting a final 'winner' which is then propagated to the A-C and implemented in the physical system.

4 EXPERIMENTS

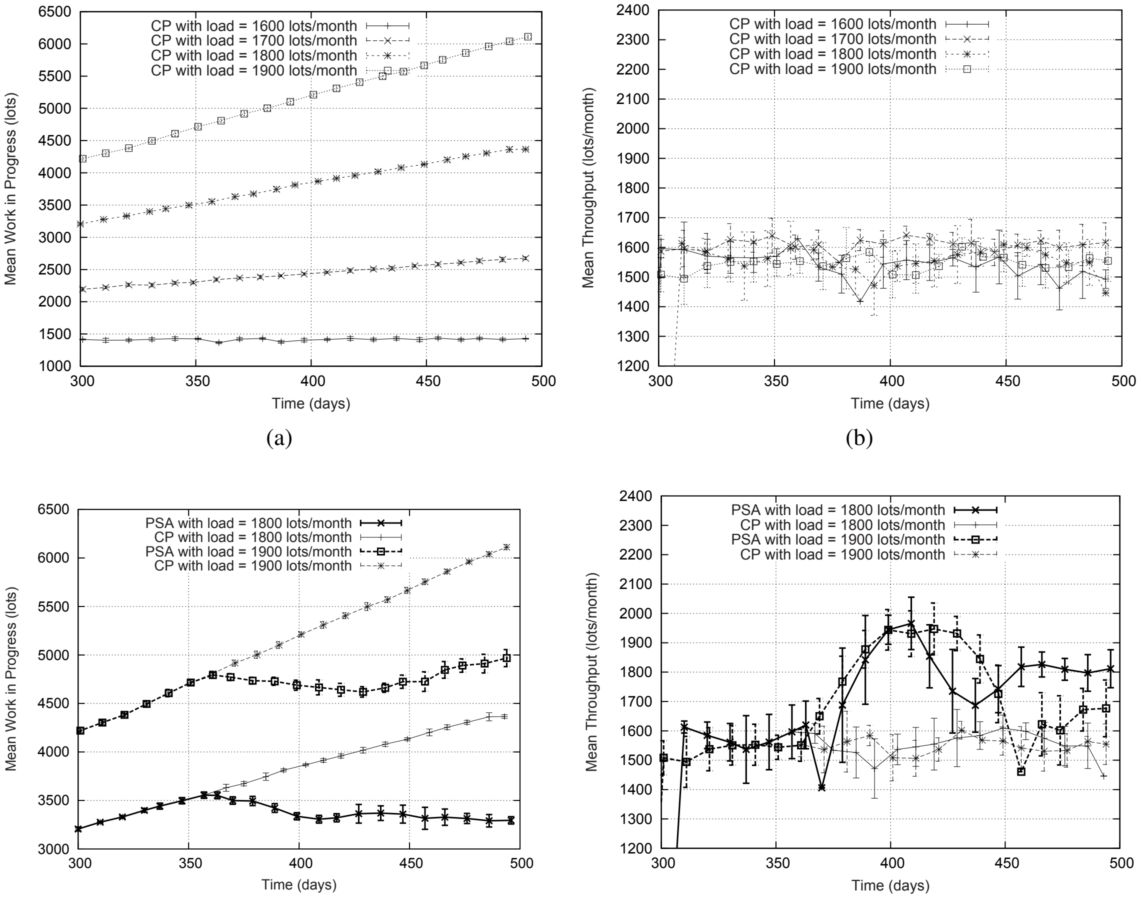
We compare the performance of the physical system when using the common practice approach for decision making (i.e., no symbiotic simulation) and the symbiotic simulation-based approach using the problem solver agent. The purpose of this comparison is to determine the maximum load the physical system can handle depending on the decision making approach used. As explained earlier in Section 3, a what-if analysis process is triggered every 12 hours and simulations of the various what-if scenarios cover the next 24 hours. Each what-if analysis is given a computing budget of 48000 simulations. In addition, each what-if scenario is evaluated using 3 simulation replications. With the given computing budget this results in a maximum of 16000 what-if scenarios that can be evaluated by each what-if analysis.

For practical reasons, experiments were not performed with a real physical system. Instead, we used a paced simulation of the fab to emulate the physical system. This enabled us to perform experiments, reflecting a period of several months, within a few weeks. We also used a mechanism to create snapshots of the complete state of the emulator at any time. These snapshots have been used as initial state for the simulation of what-if scenarios. The simulation model is deterministic with the only exception being the setup times. In practice, the exact setup times are not known *a priori*. Therefore, all simulations performed by the what-if analysis process use different random number generator seeds to produce different values for the various setup times according to a normal distributions (see Table 2 for details regarding the setup times). Both, the emulator as well as the simulations of the what-if scenarios use the model discussed in Section 2.

For evaluation purposes, we consider the work in progress (WIP) and the average throughput of the physical system. The WIP is used to determine whether the decision making approach is capable of handling a certain amount work load. More specifically, we consider work loads between 1600 and 1900 lots per month for common practice decision making and 1800 and 1900 lots per month for the problem solver agent. A set of five experiments is performed for each load and decision making approach. The results presented below represent the corresponding averages (for WIP and throughput) over these five replications. As for the conduct of experiments, for each experiment the emulator has been started from scratch, i.e., the semiconductor manufacturing system is empty with no WIP. Wafer lots are released daily according to the corresponding work load. Figure 4 shows the performance of the physical system when using the common practice approach and when using the problem solver agent. The problem solver agent is only activated at day 360. In addition, both figures only show the performance starting from day 300. The period before this day is considered the warm-up period and is thus not illustrated.

When using the common practice approach, the maximum load that can be handled by the physical system is approximately 1600 lots per month. At higher loads, the physical system cannot keep up which results in a steadily increasing WIP (see Figure 4a). The upper work load limit is also indicated by the throughput which is approximately between 1500 and 1600 lots per month regardless the load (see Figure 4b). In order to handle a given work load, the physical system needs to be able to maintain a throughput which equals the load. Clearly, this is not the case for work loads above 1600 lots per month when using the common practice approach. In contrast, when using the problem solver agent, work loads up to 1800 lots per month can be handled. This is indicated by a WIP that does not significantly increase or decrease (see Figure 4c). In addition, the problem solver agent can maintain a throughput of up to 1800 lots per month (see Figure 4d).

Initially, after activating the problem solver agent, throughput increases sharply and even exceeds the load. For example, in Figure 4d, throughput reaches almost 2000 lots per month around day 400 which



(a) (b)

(c) (d)

Figure 4: Performance of the semiconductor manufacturing system in terms of WIP (left column) and throughput (right column) when using common practice decision making only (CP, top row) and in direct comparison with the problem solver agent (PSA, bottom row).

is significantly higher than the load of 1800 or 1900 lots per month used for these experiments. This is due to the fact that, before the problem solver agent was activated, only common practice decision making has been used. Since the common practice approach cannot handle work loads of 1800 and 1900 lots per month, excess WIP has been accumulating in the queues of the physical system. This excess WIP is eventually processed when the problem solver agent is activated. As a result, when using a load of 1800 lots per month, the throughput converges to approximately the same level. However, even the problem solver agent cannot handle a load of 1900 lots per month and throughput remains significantly below this level (see Figure 4d) which explains the increasing WIP (see Figure 4c).

5 DISCUSSION

Although the problem solver agent is (in theory) capable of making decisions autonomously, it can be expected that in a real-world application scenario, the problem solver agent is not directly instructing operators but rather a senior operations manager who will take the suggestions by the problem solving agent and (if found to be appropriate) forward them to the corresponding operators. This may lead to sub-optimal decision making because the operations manager may not immediately see the advantage of implementing changes suggested by the problem solver agent. In addition, operators do not always accurately follow their instructions. The simulation model assumes that instructions by the problem solver agent are strictly followed, i.e., human operator are assumed to always follow their instructions. Although problems with human interference are an important issue, they are beyond the scope of this work and thus not further discussed. Nevertheless for a real-world application of the problem solver agent, issues of human interference need careful consideration and have to be dealt with accordingly.

The focus of this application has been an automated approach: a problem solver agent observes the physical system and triggers a what-if analysis every 12 hours. Upon triggering, the problem solver agent analyses the current situation, identifies a problem (in terms of Γ and Ω), and solves it. However, an approach in which the problem solver agent is autonomous may not be desirable in practice for various reasons. The semiconductor manufacturing process is a delicate issue and has significant impact on the profitability of the fab. A manufacturer may not be willing to leave decision making entirely up to a software system. A semi-automated approach may thus be preferred. An advantage of our approach is the availability of possible intervention points at which a user can intervene and interact with the problem solver agent. For example, a user may manually specify Γ and Ω (i.e., the problem solver agent would only be responsible for creating and evaluating the various what-if scenarios).

6 CONCLUSIONS

The problem considered in this application is concerned with decision making regarding a schedule for changing setups of tools. In contrast to our previous work (Aydt et al. 2008), we have demonstrated that symbiotic simulation can be used to optimize an entire fab rather than just a single station family. In contrast to other work in the literature, our work is not aiming at alternating different policies (e.g., such as the work by Zhang et al. (Zhang, Jiang, and Guo 2008)). Instead, we aim at an approach that allows making fine grained changes to the operations of the fab in order to improve the performance. With our approach there is no need to define specific policies for different situations. Instead, the problem solver agent is capable of scheduling setup changes directly without the need to rely on a particular scheduling rule or human intervention.

REFERENCES

Akçali, E., R. Uzsoy, D. G. Hiscock, A. L. Moser, and T. J. Teyner. 2000, December. “Alternative loading and dispatching policies for furnace operations in semiconductor manufacturing: a comparison by simulation”. In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1428–1435. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Aydt, H. 2011. *An Agent-based Framework for Problem Solving in Symbiotic Simulation Systems*. Ph. D. thesis, School of Computer Engineering, Nanyang Technological University.

Aydt, H., S. J. Turner, W. Cai, M. Y. H. Low, P. Lendermann, and B. P. Gan. 2008. “Symbiotic Simulation Control in Semiconductor Manufacturing”. In *Proceedings of the International Conference on Computational Science*, edited by M. Bubak, G. van Albada, J. Dongarra, and P. Sloot, Volume 5103, 26–35. Krakow, Poland.

Aydt, H., S. J. Turner, W. Cai, M. Y. H. Low, Y.-S. Ong, and R. Ayani. 2011. “Toward an Evolutionary Computing Modeling Language”. *IEEE Transactions on Evolutionary Computation* 15 (2): 230–247.

Bonissone, P. P., R. Subbu, N. Eklund, and T. R. Kiehl. 2006, June. “Evolutionary Algorithms + Domain Knowledge = Real-World Evolutionary Computation”. *IEEE Transactions on Evolutionary Computation* 10 (3): 256–280.

Fogel, D. 2000, February. “What is evolutionary computation?”. *IEEE Spectrum* 37 (2): 26,28–32.

Fujimoto, R., D. Lunceford, E. Page, and A. U. (editors). 2002, August. “Grand Challenges for Modeling and Simulation: Dagstuhl report”. Technical Report 350, Schloss Dagstuhl. Seminar No 02351.

Gan, B. P., L. P. Chan, and S. J. Turner. 2006, December. “Interoperating Simulations of Automatic Material Handling Systems and Manufacturing Processes”. In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1129–1135. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Potoradi, J., O. Boon, J. Fowler, M. Pfund, and S. Mason. 2002, December. “Using Simulation-based Scheduling to Maximize Demand Fulfillment in a Semiconductor Assembly Facility”. In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, 1857–1861. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Scholl, W., and J. Domaschke. 2000, August. “Implementation of Modeling and Simulation in Semiconductor Wafer Fabrication with Time Constraints Between Wet Etch and Furnace Operations”. *IEEE Transactions on Semiconductor Manufacturing* 13 (3): 273–277.

Sethi, S., K.-F. Chu, and H. Yan. 1999. “Efficient setup/dispatching policies in a semiconductor manufacturing facility”. In *Proceedings of the 38th IEEE Conference on Decision and Control*, Volume 2, 1368 – 1373.

Wu, M.-C., Y. Huang, Y. Chang, and K. Yang. 2006. “Dispatching in semiconductor fabs with machine-dedication features”. *The International Journal of Advanced Manufacturing Technology* 28:978–984.

Zhang, H., Z. Jiang, and C. Guo. 2008. “Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology”. *The International Journal of Advanced Manufacturing Technology* 41 (1–2): 110–121.

AUTHOR BIOGRAPHIES

HEIKO AYDT is a Research Associate and PhD student in the School of Computer Engineering at Nanyang Technological University (NTU) in Singapore. He received his MSc from the Royal Institute of Technology (KTH) in Stockholm. His current research interests include: Parallel and Distributed Simulation, Simulation-based Optimization, and Evolutionary Computing. His e-mail address is aydt@ntu.edu.sg.

WENTONG CAI is a Professor in the Division of Computer Science, School of Computer Engineering at Nanyang Technological University (NTU) in Singapore. He is also the Director of the Parallel and Distributed Computing Centre. He received his PhD in Computer Science from University of Exeter (UK) in 1991. His current research interests include: Parallel and Distributed Simulation, Multi-Agent Systems, and Grid and Cluster Computing. He is an associate editor of the ACM Transactions on Modeling and Computer Simulation (TOMACS), and editorial board member of the Multiagents and Grid Systems ? An International Journal. His e-mail address is aswtcai@ntu.edu.sg.

STEPHEN JOHN TURNER is Professor of Computer Science and Head of the Computer Science Division in the School of Computer Engineering at Nanyang Technological University (NTU) in Singapore. He received his MA in Mathematics and Computer Science from Cambridge University (UK) and his MSc and PhD in Computer Science from Manchester University (UK). His current research interests include: Parallel and Distributed Simulation, Grid Computing, High Performance Computing and Multi-Agent Systems. He is steering committee chair of the Principles of Advanced and Distributed Simulation (PADS) conference and an area editor for ACM Transactions on Modeling and Computer Simulation (TOMACS). His e-mail address is assjturner@ntu.edu.sg.

BOON PING GAN is the Founder and CTO of D-SIMLAB Technologies. He received his Master of Applied Science in Computer Engineering from NTU. His e-mail address is boonping@d-simlab.com.