Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

SIMULATION-BASED MULTI-MODE RESOURCE-CONSTRAINED PROJECT SCHEDULING OF SEMICONDUCTOR EQUIPMENT INSTALLATION AND QUALIFICATION

Junzilan Cheng

School of Computing, Informatics, and Decision
Systems Engineering (CIDSE)
Arizona State University
Tempe, AZ 85281, USA

John Fowler

Department of Supply Chain Management,
W. P. Carey School of Business
Arizona State University
Tempe, AZ 85287, USA

Karl Kempf

Decision Engineering Group
Intel Corporation, 5000 W. Chandler Blvd
Chandler, AZ 85226 USA

ABSTRACT

Ramping up a semiconductor wafer fabrication facility is a challenging endeavor. One of the key components of this process is to contract and schedule multiple types of resources in installing and qualifying the capital intensive and sophisticated manufacturing equipment. Due to the stochastic nature of the business environment, equipment shipment delays and activity duration increases are common. We first model the process as a deterministic multi-mode resource-constrained project scheduling problem (MRCPSP) which is NP-hard in the strong sense. Then we extend the classical MRCPSP to handle special aspects of the semiconductor environment such as time-varying resource constraints and resource vacations, alternative resource modes, non-preemptive activity splitting, etc. In this research, a modified Simulated Annealing (SA) algorithm combined with Monte Carlo simulation is proposed to evaluate and improve the execution of the Install/qual schedule with stochastic ready times and activity durations. A case study is provided to demonstrate the approach.

1 INTRODUCTION

1.1 Capital Equipment Supply Chain Overview

As a highly capital intensive industry, the investment in a semiconductor fabrication (fab) facility is extremely expensive. Nowadays, a state-of-the-art semiconductor fab costs approximately \$3-5 billion and the majority of the cost goes to the capital equipment inside the fab. It is not uncommon that a single piece of semiconductor equipment costs over ten million dollars. The capital equipment supply chain includes four major processes: equipment purchase, equipment shipment, equipment installation and qualification (Install/qual), and wafer manufacturing. When a piece of equipment completes all qualification processes, it is ready for wafer manufacturing. For a modern fab, there are generally about 2000 pieces of capital equipment that need to go through this supply chain to be ready for wafer manufacturing.

1.2 The Install/Qual Process

The Install/Qual scheduling is the process of ramping up a fabrication facility which includes three serial activities: Physical Installation, Supplier Qualification and Company Qualification. Physical installation is the process of installing the infrastructure needed by the equipment (e.g. pipes for water and gases) and then physically installing the equipment. Physical installation is conducted by trades (e.g. architects, electricians, mechanics, and plumbers) who are contracted from outside the company. Supplier qualification is the qualifying process carried out by equipment suppliers (supplier resource). Since equipment may be supplied by different suppliers, there are multiple supplier resources. Finally, the company has its own qualifying engineers (company resource) responsible for the company qualification process.

The Install/Qual process takes almost half of the supply chain lead time (time to first good wafer produced) and plays an important role in the efficiency of the entire supply chain. The velocity of ramping up a wafer fab can impact a semiconductor company's bottom line by determining how fast the next generation product can be made available to customers. Overestimation of future demand leads to idle capital equipment which could be millions of dollars wasted, while underestimation of future demand could end up with lost sales and even larger profit loss. Thus, a better coordinated Install/Qual schedule can shorten the capital equipment supply chain lead time and delay decision-making which can diminish the variance in planned factory capacity based on predicted future customer demand.

Practical aspects in the Install/Qual process make the scheduling problem a challenging endeavor. Multiple resources (human resources, project budget, factory floor space, etc) are involved in processing activities and an activity might have multiple processing options. For example, a piece of equipment can be installed by 3 senior and 1 junior technicians with a total cost of $20K in 6 working days or 1 senior and 3 junior technicians with a total cost of $16K in 8 working days.

Due to the stochastic nature of the business environment, there is a significant amount of uncertainty associated with the activity ready times and durations in the Install/Qual process. During the preliminary planning stage of the Install/Qual process, the uncertainty is particularly large for new technologies since the scope and obstacles to the project are still undefined.

The Install/Qual scheduling phase of the practical example studied in this research mostly relies on information from previous Install/Qual processes (this often referred to as tribal knowledge) and the reschedule phase is generally based on manual inspection. It is an inefficient and time-consuming process which has a high probability of producing a poor schedule. In this research, the Install/Qual scheduling problem is mathematically formulated as a MRCPSP model. A modified Simulated Annealing (SA) algorithm combined with Monte Carlo simulation provides an approach to evaluate and improve the Install/Qual scheduling in an uncertain business environment.

2 PROBLEM STATEMENT

MRCPSPs (multi-mode resource-constrained project scheduling problems) are widely used in many real-world project management applications. A project network $G(N, A)$ using the AoN (activity on node) convention contains a set of nodes N representing the activity set $\mathbb{N}$ ($|\mathbb{N}| = n$) and a set of directed arcs A representing the precedence relations among activities. An activity can only start when its predecessors are finished. For the purpose of network completeness, a dummy start node 0 and a dummy finish node $n + 1$ are added into the network. Within the discussion in this paper, if not otherwise stated, we treat "activities", "tasks" and "jobs" interchangeably. Mathematical notation is provided in Table 1.

Both renewable resources $\mathbb{R}^r$ and non-renewable resources $\mathbb{R}^n$ are considered in this work. The availability of a renewable resource k ($k \in \mathbb{R}^r$) is restricted in each time period t. Examples of renewable resources are the number of skilled technicians available per day and the number of testing equipment available per shift. The availability of a non-renewable resource k ($k \in \mathbb{R}^n$) is restricted throughout the whole planning horizon $[0, T]$. Examples of non-renewable resources include the total budget for the entire project, the total available factory floor space, and the total available amount of raw materials. For renewable resources, the "resource profile" function specifies the availability of a particular resource over time. Each activity j ($j \in \mathbb{N}$) has multiple processing modes Mod_j to choose from and each mode $m \in Mod_j$ has a corresponding activity duration p_j^m and consumes r_{jk}^m amount of resource k.

In practice, the ready time rad_j of machine j indicates when the equipment arrives at the factory site. This time has variability due to the uncertainty in the capital equipment supply chain, e.g. shipment delays. Activity duration p_j^m is also uncertain since the execution of an activity depends on the estimation of work content which cannot be guaranteed to be completely accurate. The uncertainty of an activity duration is particularly large on new technologies where no historical data on the workload exists. Thus, both activity ready time rad_j and duration p_j^m are considered as random variables where their distributions are known or estimable.

Table 1: Mathematical Notation

Symbol	Description
j	An activity/task/job j
t	Time t
k	Resource type
m	Activity processing mode
Mod_j	Set of available processing modes for activity j
p_j^m	Processing duration of activity j under mode m
rad_j	Ready time of activity j
due_j	Due date of activity j
r_{jk}^m	Required amount of resource type k on activity j under mode m
U_k	Upper bound on availability for resource type k
L_k	Lower bound on availability for resource type k
U_{kt}	Upper bound on availability for resource type k at time t
L_{kt}	Lower bound on availability for resource type k at time t
$\mathbb{N}$	Activities in the project $\mathbb{N} = \{1, 2, \dots, n\}$
$\mathbb{R}^r$	Set of renewable resources
$\mathbb{R}^n$	Set of non-renewable resources
T	Maximum project planning horizon
$G(N, A)$	Network G - N represents nodes and A represents arcs
$arc(i, j)$	Directed arc connecting node i to node j
$pred(j)$	The set of predecessor activities of activity j. $pred(0) = \emptyset$
$succ(j)$	The set of successor activities of activity j. $succ(\lvert\mathbb{N}\rvert + 1) = \emptyset$

In the Install/Qual scheduling problem, only non-preemptive activity splitting is allowed (Cheng et al., 2012) which indicates that activities can only split when renewable resources are not available (weekends, holidays) or there is less than the required amount. Intuitively, MRCPSP with non-preemptive activity splitting can be considered as the intermediate case between MRCPSP and preemptive MRCPSP (Buddhakulsomsiri and Kim 2006).

Decision variables are: $y_j^m = 1$ if activity $j \in \mathbb{N}$ is being processed in mode $m \in Mod_j$ and 0 otherwise; $x_{jt}^m = 1$ if activity $j \in \mathbb{N}$ is being processed in mode $m \in Mod_j$ at time $t = 1, 2, \dots, T$ and 0 otherwise; resulting variables S_j and C_j represent the start time and completion time of activity j, respectively. $S_{|\mathbb{N}|+1}$ is the start time for the dummy finish activity which is also the makespan of the project.

An indicator function is introduced to specify whether an activity j in mode m is feasible to process at a certain time period:

$$1_{[L_{kt}, U_{kt}]}: \gamma_{jkt}^m \to \{0, 1\} \tag{1}$$

which is equivalent to:

$$1_{[L_{kt},U_{kt}]} = \begin{cases} 1 & \text{if } \gamma_{jkt}^{m} \in [L_{kt}, U_{kt}] \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Additional decision variables o_{jt} and q_{jt} are defined to indicate whether a time period t is between the start time S_j and the completion time C_j of activity j.

$$o_{jt} = \begin{cases} 1 & \text{if } t \le C_j \\ 0 & \text{otherwise} \end{cases}, \quad \forall j \in \mathbb{N} \tag{3}$$

$$q_{jt} = \begin{cases} 1 & \text{if } t \ge S_j \\ 0 & \text{otherwise} \end{cases}, \quad \forall j \in \mathbb{N} \tag{4}$$

Data inputs are resource profiles $[L_{kt}, U_{kt}]$ for renewable resources and $[L_k, U_k]$ for non-renewable resources. The mixed-integer programming formulation is provided as follows.

$$\min \ S_{|\mathbb{N}|+1} \tag{5}$$

Subject to:

$$\textstyle\sum_{m \in Mod_j} y_j^m = 1, \ \ \forall j \in \mathbb{N} \tag{6}$$

$$\textstyle\sum_{t=1}^{T} x_{jt}^m = p_j^m \cdot y_j^m, \quad \forall j \in \mathbb{N}, m \in Mod_j \tag{7}$$

$$C_i \le S_j - 1, \ \ \forall (i,j) \in A \tag{8}$$

$$S_j \le x_{jt}^m \cdot t + M\left(1 - x_{jt}^m\right), \ \ \forall j \in \mathbb{N}, m \in Mod_j, t = 1,2,\dots,T \tag{9}$$

$$C_j \ge x_{jt}^m \cdot t, \quad \forall j \in \mathbb{N}, m \in Mod_j, t = 1,2,\dots,T \tag{10}$$

$$S_j \ge rad_j, \quad \forall j \in \mathbb{N} \tag{11}$$

$$C_j \le due_j, \quad \forall j \in \mathbb{N} \tag{12}$$

$$L_{kt} \le \textstyle\sum_{j \in \mathbb{N}} \sum_{m \in Mod_j} r_{jk}^m \cdot x_{jt}^m \le U_{kt}, \quad \forall k \in \mathbb{R}^r, t = 1,2,\dots,T \tag{13}$$

$$L_k \le \textstyle\sum_{j \in \mathbb{N}} \sum_{m \in Mod_j} r_{jk}^m \cdot y_j^m \le U_k, \quad \forall k \in \mathbb{R}^n \tag{14}$$

$$M \cdot o_{jt} \ge C_j - t + 1, \quad \forall j \in \mathbb{N}, t = 1,2,\dots T \tag{15}$$

$$M \cdot \left(1 - o_{jt}\right) \ge t - C_j, \quad \forall j \in \mathbb{N}, t = 1,2,\dots T \tag{16}$$

$$M \cdot q_{jt} \ge t - S_j + 1, \quad \forall j \in \mathbb{N}, t = 1,2,\dots T \tag{17}$$

$$M \cdot \left(1 - q_{jt}\right) \ge S_j - t, \quad \forall j \in \mathbb{N}, t = 1,2,\dots T \tag{18}$$

$$o_{jt} + q_{jt} \ge 1, \quad \forall j \in \mathbb{N}, t = 1,2,\dots,T \tag{19}$$

$$x_{jt}^m \ge y_j^m + \gamma_{jkt}^m + o_{jt} + q_{jt} - 3, \ \ \forall j \in \mathbb{N}, m \in Mod_j, \forall k \in \mathbb{R}^n, t = 1,2,\dots T \tag{20}$$

$$4 \cdot x_{jt}^m \le y_j^m + \gamma_{jkt}^m + o_{jt} + q_{jt}, \quad \forall j \in \mathbb{N}, m \in Mod_j, \forall k \in \mathbb{R}^n, t = 1,2,\dots T \tag{21}$$

$$S_j \ge 0, \quad \forall j \in \mathbb{N} \tag{22}$$

$$C_j \ge 0, \quad \forall j \in \mathbb{N} \tag{23}$$

$$y_j^m \in \{0,1\}, \ \forall j \in \mathbb{N}, m \in Mod_j \tag{24}$$

$$x_{jt}^m \in \{0,1\}, \quad \forall j \in \mathbb{N}, m \in Mod_j, t = 1,2,\dots,T \tag{25}$$

$$o_{jt} \in \{0,1\}, \quad \forall j \in \mathbb{N}, t = 1,2,\dots T \tag{26}$$

$$q_{jt} \in \{0,1\}, \quad \forall j \in \mathbb{N}, t = 1,2,\dots T \tag{27}$$

The objective function (5) minimizes the project makespan. Constraint set (6) ensures only one mode can be selected for each activity. Constraint set (7) ensures that if mode m is selected for activity j, the total processing time must equal the corresponding duration. Constraint sets (8) – (10) are precedence constraints. Constraint sets (11) – (12) ensure ready times and due dates (in fact, deadlines) are not violated. Constraint sets (13) – (14) ensure resource availability for both renewable resources and non-renewable resources. Constraint sets (15) – (18) are included to support the new decision variables o_{jt} and q_{jt}. Constraint set (19) ensures the activity completion time is no earlier than the start time for activity j. Constraint sets (20) – (21) ensure an activity j cannot be preempted at time t if it is eligible. Constraint sets (22) – (27) are the non-negativity and binary constraints. A big number M in the MIP formulation is set to be the maximum project planning horizon T.

3 METHODOLOGY

Literature related to RCPSP dates back to 1950's with the development of PERT (program evaluation and review technique, cf. Malcolm et al. (1959)) and CPM (critical path method, cf. Kelley (1963)). Cheng et al. (2012) summarized major research extensions to RCPSP and pointed out that MRCPSP is one of the most adopted extensions. Early review papers on RCPSP can be found in Davis (1973), Icmeli et al. (1993), Özdamar and Ulusoy (1995), Herroelen et al. (1998), Brucker et al. (1999) and Kolisch and Padman (2001). More recent ones include Hartmann and Briskorn (2010) and Węglarz et al. (2011).

The MRCPSP problem is strongly NP-hard since it is the generalization of the well-known job shop scheduling problem (Blazewicz et al. 1983). For MRCPSP with more than one non-renewable resource, the problem of finding a feasible solution is NP-complete (Kolisch and Drexl, 1997). As pointed out in Węglarz et al., 2011), it is still computationally intractable to find optimal solutions for MRCPSP instances with more than 20 activities and 3 modes per activity.

Thus, heuristic approaches are considered in most research efforts in MRCPSP (Węglarz et al., 2011) research and the project makespan is generally considered as the problem objective. However, a deterministic schedule makespan cannot evaluate the "goodness" of executing a scheduling in the presence of stochasticity. Schedules with the same deterministic makespan might perform differently when stochastic elements are introduced.

The application of simulation in RCPSP is surprisingly rare. Golenko-Ginzburg and Gonik (1997) develop a heuristic procedure for RCPSP with stochastic activity durations with the objective of minimizing the expected project duration. A simulation approach is adopted to approximate the probability of each activity being in the critical path of the project. Lee (2005) introduces a Monte Carlo simulation-based software to measure the probability to complete a project in a certain time when activity durations follow given probability density functions. Pappert et al. (2010) proposed a framework for simulation-based scheduling to solve planning and scheduling assembly line problems.

In this research, a SA combined with a Monte Carlo simulation approach is proposed to evaluate different schedules and search for better schedules when there are stochastic activity durations and ready times. The main reason for choosing SA over other meta-heuristics is that SA performs well in local search and the "temperature" parameter is easy to implement as a threshold to determine when to start simulating candidate solutions. Other meta-heuristics can be studied as potential future research.

Figure 1 illustrates the basic framework of the simulation-based scheduling approach. After creating an initial Install/Qual schedule, the simulation module simulates the schedule based on probability distributions of activity durations and ready times and analyzes the average and standard deviation of the simulation results. If the schedule and simulation results cannot satisfy the requirements of senior management, they will be sent to the optimization module to improve the schedule. A new schedule will be generated and sent back to the simulation module after feasibility checks regarding resource constraints. Since the size of the Install/Qual scheduling problem instance exceeds common academic RCPSP problem instances, only heuristics will be considered when designing the optimization module.

The pseudo code for our SA is provided in the Appendix. Since the performance of most search procedure is known to be dependent upon initial solutions, the initial solution in this work is not generated in a completely random manner: first, a mode improvement procedure in (Hartmann 2001) is incorporated as a local search procedure that modifies the mode assignments for infeasible solutions (infeasible with respect to non-renewable resources). It is worth mentioning that since it is NP-complete to find a feasible schedule in MRCPSP (Kolisch and Sprecher 1997), there is no guarantee that the Hartmann (2001) search can find a "feasible" schedule with respect to non-renewable resources. Second, the SA runs multiple times each with different initial solutions and the best run is selected. Details of the mode improvement and the SA procedure can be found in the Appendix.

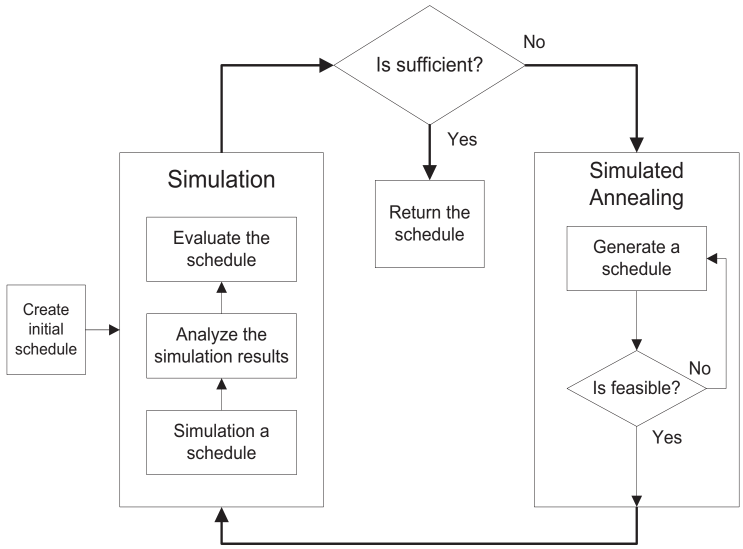

Figure 1: Flow chart of the simulation-based scheduling

Schedules are represented with a random key (RK) representation in which the first random key vector represents the relative sequence priority and the second random key vector represents the mode assignment. SA with RK representation in RCPSP can be found in (Lee and Kim 1996) and (Cho and Kim 1997). Debels and Vanhoucke (2007) illustrated that the RK representation leads to promising results in RCPSP if topological ordering (TO) (Valls et al. 1999) is applied. The TO of activities is an order which is compatible with precedence relations of the projects. It implies that for all activities i and j for which the starting time of activity i is earlier than the starting time of activity j, activity i should have a higher priority than activity j. Another advantage of the RK representation is that it always maintains precedence feasibility. Compared to parallel schedule generation scheme (SGS), serial SGS (Sprecher et al. 1995, Kolisch 1996) is used in this work since parallel SGS might not always include the optimal solution.

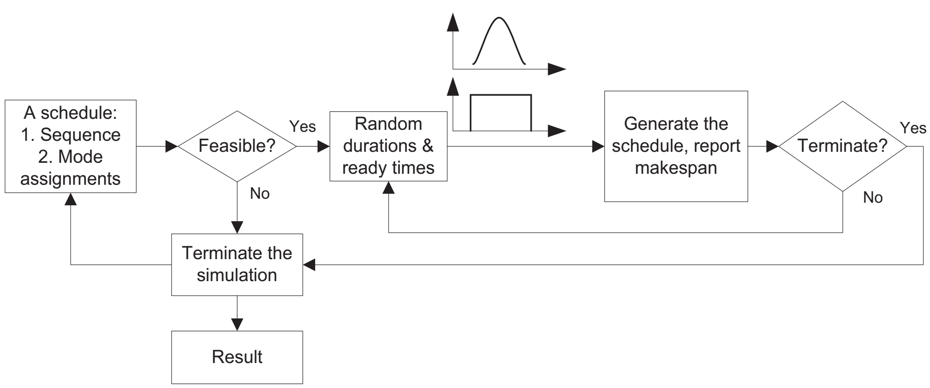

Figure 2: Flow chart of the Monte Carlo simulation module

As described above, a solution in the SA algorithm is encoded as two strings of number representing priorities of activities and mode assignment. The neighborhood of a solution can be obtained from the current solution through an interchange method (Lee and Kim 1996): two activities are selected and the priorities (RKs) and mode assignments are exchanged. In this method, the first activity j is selected randomly and the other activity is selected randomly among activities whose indexes are between $[j-\alpha\beta, j+\alpha\beta]$ where β is the maximum number of predecessors or successors of an activity in the project network, and α is a parameter determined by preliminary experiments. This can limit the neighborhood search on activities that are highly likely to compete for resources and prevent searching for unnecessary alternatives. Basic SA parameters such as initial temperature, cooling ratio, epoch length, etc. are set by a series of preliminary experiments. The Monte Carlo simulation module generates activity ready times and durations following specified distributions. The simulation procedure is illustrated in Figure 2.

4 CASE STUDY

A project scheduling case study with 15 activities is provided to demonstrate the simulation-based scheduling approach. The real Install/Qual dataset is not tested in this work for confidentiality reasons and the extreme size of the real-world dataset. The problem instance is from the "c1537_1" instance in the set "C15" of MRCPSP instance sets from PSPLIB (http://129.187.106.231/psplib/). In order to represent a mini Install/Qual scheduling problem, the problem instance is modified by considering time-varying resource profiles, random resource vacations, multiple alternative processing modes and non-preemptive activity splitting. The detailed problem instance modification procedures can be found in Cheng et al. (2012). More benchmark problem instances and ultimately the real-world dataset will be studied in future research. The project network is shown in Figure 3 with the activity-on-node (AON) representation. Activities 1 and 18 are dummy activities with 0 activity durations and no resource requirements. There are two types of renewable resources and two types of non-renewable resources. Each activity has three alternative processing modes and activity splitting is not allowed.

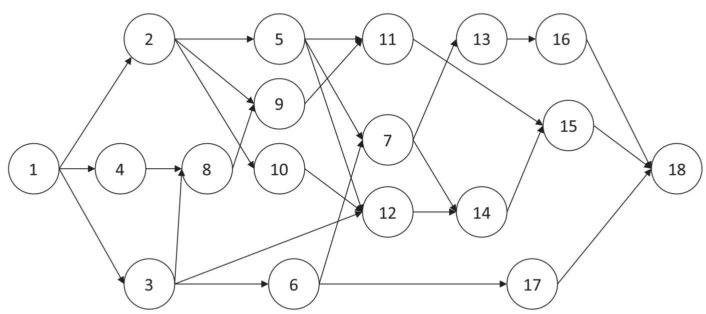

Figure 3: Project network

Resource profiles for renewable resource R1 and renewable resource R2 are provided in Figure 4. The resource profiles are modified from the deterministic resource limits in the original problem instance by introducing time-varying resource limits and random resource vacations. The resource limits drop to zero indicating resource vacations. Detailed procedures can be found in Cheng et al. (2012).

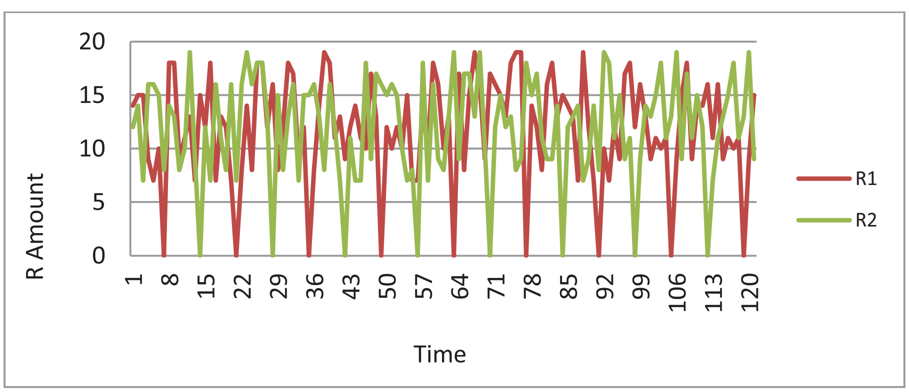

Figure 4: Resource Profile

In the current model, activity ready times are normally distributed while activity durations are uniformly distributed. Other potential distributions can be easily incorporated into the model as input data. 1000 runs for each simulation are selected through preliminary experiments and the runtime for 1000 runs is less than 1 sec. The output for the simulation is the average makespan. Other statistical information such as the variance and confidence intervals are also analyzed. The simulation model is programmed in Visual Studio C++ 2005 Edition on a desktop with an Intel® 2 Quad Core™ CPU Q9400 @ 2.66GHz, 4.00 GB installed memory, and the Windows 7 Enterprise 64-bit Operating System.

The results are presented in the following two figures. *D_Current* represents the current deterministic makespan while *D_Best* represents the current best deterministic makespan. Since SA accepts many "bad" solutions in the early stage of the cooling process, infeasible solutions regarding non-renewable resources might also be accepted to maintain solution diversity and avoid local optima.

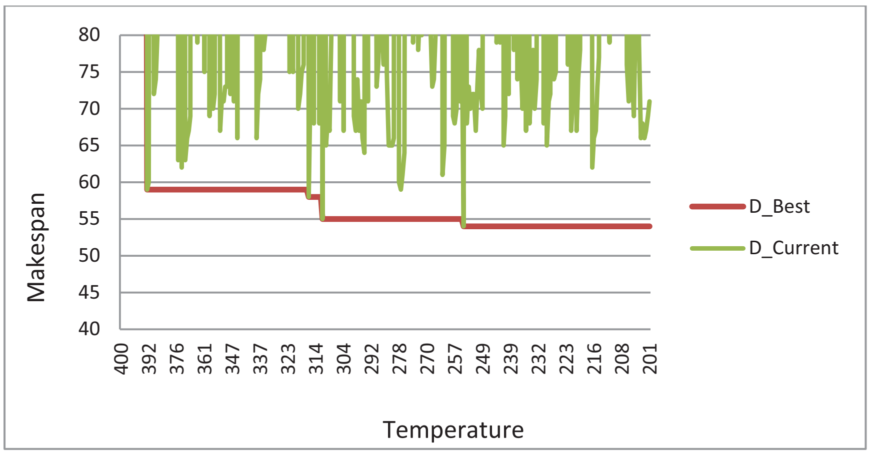

Figure 5: Deterministic Makespan found by SA

In Figure 5, the results indicate the stage of SA where simulation is used has not yet been triggered. The best deterministic makespan found is 54 when the temperature is 200.

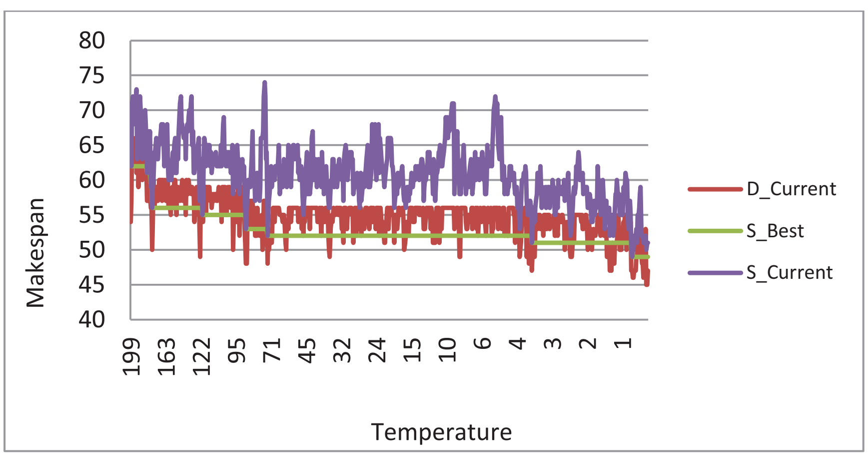

Figure 6: Stochastic Makespan found by SA

The best deterministic schedule found in the first stage is considered as the starting point for the simulation process. *D_Best* represents the best deterministic makespan, *S_Best* represents the best average makespan and *S_Current* represents the average makespan for the current schedule. Average values are rounded up to the nearest integer. Until the freezing temperature, the best average makespan is 49 with the corresponding deterministic makespan 47 (Figure 6).

In Table 1, a partial set of the "good" schedules including the "best" schedule found so far regarding stochastic makespan is presented. An interesting observation is that the schedule with the smallest *S_Current* (schedule 6) is not necessarily the schedule with the smallest *D_Current* (schedule 12). Thus, even though schedules with small deterministic makespans tend to have small stochastic makespans, simply relying on the deterministic makespan does not guarantee the best performance when there are stochastic activity durations and ready times.

Table 1: Partial results

Schedule	Temp	D_Current	S_Current	Confidence Interval
1	1.11854	48	52.246	[51.951, 52.541]
2	1.08532	47	51.957	[51.662, 52.252]
3	0.91486	46	49.894	[49.602, 50.186]
4	0.91486	47	51.617	[51.299, 51.934]
5	0.81911	47	51.883	[51.575, 52.191]
6	**0.69744**	**47**	**48.144**	**[47.875, 48.413]**
7	0.69744	47	51.373	[51.039, 51.707]
8	0.59384	47	50.802	[50.498, 51.106]
9	0.52637	46	50.223	[49.936, 50.510]
10	0.49557	46	50.704	[50.408, 51.000]
11	0.41355	46	49.708	[49.425, 49.991]
12	0.40127	45	50.590	[50.292, 50.884]
13	0.30285	46	49.763	[49.471, 50.055]
14	0.01076	45	49.891	[49.619, 50.163]

5 CONCLUSIONS

In this research, the Install/Qual process in semiconductor fab facility is modeled as a multi-mode resource constrained project scheduling problem with non-preemptive activity splitting. A SA combined with a Monte Carlo simulation approach is proposed to evaluate different schedules where activity ready times and durations are stochastic. A case study illustrates the model and demonstrates the problem solving approach. As expected, the results also demonstrate that by simply solving as a deterministic MRCPSP problem, the solution found might not be the best schedule in a stochastic environment. For possible future research, other distributions for activity ready times and durations will be considered. More problem instances must be tested and analyzed as well.

APPENDIX

Mode Improvement Procedure

Step 1: Randomly select an activity $j \in J$ and its current mode assignment is y_j^m

Step 2: Pick a new mode assignment $\hat{y}_j^m$ for j

Step 3: If $\sum_{m \in Mod_j} r_{jk}^m \cdot \hat{y}_j^m \leq \sum_{m \in Mod_j} r_{jk}^m \cdot y_j^m, \forall k \in \mathbb{R}^n$, the new mode assignment $\hat{y}_j^m$ is selected to replace the old mode assignment y_j^m

Step 4: Return to step 1 for *Num_Repeat* times or until found a feasible mode assignment

SA Algorithm

```
Step 1: Initialization
        sch_initial ← random
        sch_current ← sch_initial
        sch_best ← sch_initial
        D_makespan_current ← D_makespan(sch_initial): serial SGS
        D_makespan_best ← D_makespan_current
        Temperature_current ← Temperature_initial
        Temperature_freeze ← A predefined freezing temperature
        Threshold ← the criteria that determines when to run the simulation
        epoch_length, the number of neighbour search iterations at a temperature level
Step2: Cooling process (Temperature_current >= Temperature_freeze)
        Step 2-1:
        sch_new = neightbour(sch_current): interchange method
        While (Temperature_current >= Threshold * Temperature_initial)
            D_makespan_new = D_makespan(sch_new)
            If (D_makespan_new <= D_makespan_current)
                Accept sch_new, move to the neighbour
                update sch_current, update D_makespan
                If D_makespan_new <= D_makespan_best
                    update D_makespan_best
                    update sch_best
            If (D_makespan_new > D_makespan_current)
                If rand() <= e^(-Δ/T), accept the neighbor
                Δ: D_makespan_new - D_makespan_current
                T: Temperature_current
                        update sch_current, update D_makespan
        While (Temperature_current < Threshold * Temperature_initial)
            S_makespan_new = S_makespan(sch_new): simulate Num_sim runs, return the avg. makespan
            If (S_makespan_new <= S_makespan_current)
                Accept sch_new, move to the neighbor
                update sch_current, update S_makespan
      If (S_makespan_new <= S_makespan_best)
                        update S_makespan_best, sch_best
                        if (S_makespan_new > S_makespan_current)
                               if rand() <= e^(-Δ/T), accept the neighbor
                               Δ: S_makespan_new - S_makespan_current
                                     update sch_current, update S_makespan
         Step 2-2:
         repeat step 2-1 for epoch_length times
Step3: Update temperature T = cT, in which c is the cooling ratio between [0, 1]. Go back to step 2
```

Simulation Procedure

Step 1: Generate random activity durations and ready times according to the specified distributions
Step 2: Generate the schedule using the serial SGS, return the makespan
Step 3: Repeat step 2 for *Num_Sim* (number of simulation replications) times, calculate the average makespan

REFERENCES

Blazewicz, J., Lenstra, J. K., and Rinnooy Kan, A. H. G. 1983. "Scheduling Subject to Resource Constraints: Classification and Complexity". *Discrete Applied Mathematics* 5: 11-24.

Brucker, P., Drexl, A., Moehring, R., Neumann, K., and Pesch, E. 1999. "A Resource-constrained Project Scheduling: Notation, Classification, Models, and Methods". *European Journal of Operational Research* 112: 3-41.

Buddhakulsomsiri, J., and Kim, D. S. 2006. "Properties of Multi-mode Resource-constrained Project Scheduling Problems with Resource Vacations and Activity Splitting". *European Journal of Operational Research* 175: 279-295.

Cheng, J., Fowler, J., Kempf, K., and Mason, S. 2012. "Multi-mode Resource-constrained Project Scheduling Problems with Non-preemptive Activity Splitting". Working Paper.

Cho, J., and Kim, Y. 1997. "A Simulated Annealing Algorithm for Resource-constrained Project Scheduling Problems". *Journal of the Operational Research Society* 48: 736-744.

Davis, E. W. 1973. "Project Scheduling Under Resource Constraints-historical Review and Categorization of Procedures". *IIE Transactions* 5: 297-313.

Debels, D., and Vanhoucke, M. 2007. "A Decomposition-based Genetic Algorithm for the Resource-constrained Project Scheduling Problem". *Operations Research* 55: 457-469.

Golenko-Ginzburg, D., and Gonik, A. 1997. "Stochastic Network Project Scheduling with Non-consumable Limited Resources". *International Journal of Production Economics* 48: 29-37.

Hartmann, S. 2001. "Project Scheduling with Multiple Modes: A Genetic Algorithm". *Annals of Operations Research* 102: 111-135.

Hartmann, S., and Briskorn, D. 2010. "A Survey of Variants and Extensions of the Resource-constrained Project Scheduling Problem". *European Journal of Operational Research* 207: 1-14.

Herroelen, W., De Reyck, B., and Demeulemeester, E. 1998. Resource-constrained Project Scheduling: a Survey of Recent Developments. *Computers & Operations Research*, 25, 279-302.

Icmeli, O., Erenguc, S. S., and Zappe, C. J. 1993. "Project Scheduling Problems: a Survey". *International Journal of Operations & Production Management* 13: 80-91.

Kelley, J. E. J. 1963. *The Critical Path Method: Resource Planning and Scheduling*. New Jersey: Prentice Hall.

Kolisch, R., and Padman, R. 2001. "An Integrated Survey of Deterministic Project Scheduling". *Omega* 29: 249-272.

Kolisch, R., and Sprecher, A. 1997. "PSPLIB-A Project Scheduling Problem Library* 1:: OR Software-ORSEP Operations Research Software Exchange Program". *European Journal of Operational Research* 96: 205-216.

Kolisch, R., and Drexl, A. 1997. "Local Search for Nonpreemptive Multi-mode Resource-constrained Project Scheduling". *IIE Transactions* 29: 987-999.

Lee, D. 2005. "Probability of Project Completion Using Stochastic Project Scheduling Simulation". *Journal of Construction Engineering & Management* 131: 310-318.

Lee, J., and Kim, Y. 1996. "Search Heuristics for Resource-constrained Project Scheduling". *The Journal of the Operational Research Society* 47: 678-689.

Malcolm, D. G., Roseboom, J. H., Clark, C. E., and Fazar, W. 1959. "Application of a Technique for Research and Development Program Evaluation". *Operations research* 7: 646-669.

Özdamar, L., and Ulusoy, G. 1995. "A Survey on the Resource-constrained Project Scheduling Problem". *IIE Transactions* 27: 574-586.

Pappert, F., Angelidis, E. and Rose, O., 2010. "Framework for Simulation Based Scheduling of Assembly Lines". In *Proceedings of the 2010 Winter Simulation Conference*, 1690-1698.

Valls, V., Laguna, M., Lino, P., Pérez, A., and Quinatanilla, S. 1999. "Project Scheduling with Stochastic Activity Interruptions". *International Series in Operations Research and Management Science*: 333-354.

Węglarz, J., Józefowska, J., Mika, M., and Waligóra, G. 2011. "Project Scheduling with Finite or Infinite Number of Activity Processing Modes–A survey". *European Journal of Operational Research* 208: 177-205.

AUTHOR BIOGRAPHIES

JUNZILAN "FELIX" CHENG is a PhD candidate of Industrial Engineering in the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University. His research interests include applied operations research, scheduling, project management and discrete event simulation. He received his B.S. in Industrial Engineering from Shanghai Jiao Tong University, Shanghai, China. His email address is felix.cheng@asu.edu.

JOHN FOWLER is the Motorola Professor and Chair of the Supply Chain Management department at ASU. He is also a Professor of Industrial Engineering (IE) and was previously the program chair for IE at ASU. Professor Fowler's research interests include discrete event simulation, deterministic scheduling, and multi-criteria decision making. He has published over 85 journal articles and over 100 conference papers. He was the Program Chair for the 2008 Industrial Engineering Research Conference and the 2008 Winter Simulation Conference (WSC). He is currently serving as Editor-in-Chief for a new Institute of Industrial Engineers journal focused on health care delivery systems entitled *IIE Transactions on Healthcare Systems Engineering*. He is also an Editor of the *Journal of Simulation*. He is a Fellow of the Institute of Industrial Engineers (IIE) and currently serves as the IIE Vice President for Continuing Education, is a former INFORMS Vice President, and is an SCS representative on the Winter Simulation Conference Board of Directors. His email address is john.fowler@asu.edu.

KARL KEMPF is a member of the National Academy of Engineering, an Intel Fellow and an Adjunct Professor at Arizona State University. His research interests span the optimization of production and supply chain planning and execution in semiconductor supply chains including various forms of simulation. He can be contacted by e-mail at karl.g.kempf@intel.com.