*Proceedings of the 2018 Winter Simulation Conference*
*M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.*

# OPTIMIZATION VERSUS CONSTRUCTION OF TRANSPORT SCHEDULES TO REDUCE TRAVEL TIME VARIABILITY AND AVOID CONGESTIONS IN CONVEYOR-BASED AMHS FOR WAFER FABS

Clemens Schwenke
Sebastian Jannasch
Klaus Kabitzsch

Department of Applied Computer Science
Dresden University of Technology (TUD)
01062 Dresden, GERMANY

## ABSTRACT

Advanced transport scheduling for conveyor-based automated material handling systems (AMHS) in semiconductor fabricating facilities (wafer fabs) can reduce transport times and thus cycle times. Commonly, transport operations of arriving wafer lots at conveyor junctions, such as rotary tables, are sequenced ad-hoc by myopic first-come-first-serve policies. In contrast, improved transport schedules for the transport operations can be produced ahead of the time in conjunction with the overall production schedule for process operations. More precisely, such a transport schedule can either be sequentially constructed by fixing one transport after another, or it can be optimized by simultaneously fixing several conflicting transports. Hence, two conceptually different transport scheduling methods, which both avoid congestions by enforcing a no-wait constraint, are compared with special regard to transport-related variability. Furthermore, three different AMHS models that exhibit the typical interbay-intrabay layout are used for computational experiments.

## 1 INTRODUCTION

In semiconductor manufacturing, automated material handling systems (AMHS) move the wafer lots from one workstation or machine to another, where the wafer lots undergo process operations. Modern AMHS fully automatically transport the wafer lots on particular travel paths between the machines. Hence, when creating a process operation schedule for the machines, path-dependent travel times need to be considered. In most wafer fabricating facilities (fabs), such travel times are estimated either by deriving the minimal, ideal, transport time from the length of a path or by averaging observed transport times on a path. Therefore, minimal travel times are also called raw transport times.

Unfortunately, transport time estimates cannot always be adhered to in reality. Instead, transports take longer and wafer lots arrive delayed at their destination machines. As a result, subsequent process operations will not be executed in time and the corresponding machines will be idle. Such delays can propagate within a fab's overall process operation schedule (fab schedule), so that not just one, but several wafer lots exhibit longer cycle times. In summary, wafer lots do not meet their due dates and exit the fab delayed.

The reason for transport delays are transient congestions, which only occur at highly frequented intersections and points in time of high traffic loads, e.g., when several batch machines finish process operations and then release many wafer lots simultaneously into the AMHS. Therefore, it is necessary to accurately model the transport system's behavior if the transport times shall be estimated more exactly. More specifically, the exact progression of wafer lots waiting for other wafer lots needs to be modeled. The reason being, AMHS resources are scarce, which creates bottlenecks at conveyor junctions. As a result,

978-1-5386-6572-5/18/$31.00 ©2018 IEEE 

congestion effects can be precisely simulated. But most importantly, the succession of transports can be optimized so that delays can be avoided or reduced.

Consequently, we suggest two methods for scheduling transports within conveyor-based AMHS in wafer fabs. The first method successively fills an overall transport schedule by inserting one complete wafer lot transport after another. Thus, it is called *Transport Insertion* (*TI*) method. The second method considers several possibly interfering transports at once and optimizes their sequence of passing bottleneck elements of the AMHS so that transport-related delays will be reduced. Thus, it is called *Transport Optimization* (*TO*) method. In contrast to our previous conceptual work on this matter (Schwenke and Kabitzsch 2017), we now examine the two novel methods more thoroughly as follows: Firstly, three different AMHS layouts are used for the experiments. The two new AMHS models are more generic than the previously used one. Secondly, the impact on transport-related variability in cycle times is quantified. Thirdly, the briefly introduced *TI* method is now described in greater detail. Particularly, the conceptual difference to the *TO* method is highlighted by a manageable illustrative example.

The remainder of this document is structured as follows. Section 2 highlights related work. Section 3 describes the differences of three AMHS models and the two transports scheduling methods. Accordingly, Section 4 validates the two transport scheduling methods using the three described AMHS models. Finally, Section 5 draws conclusions.

## 2 RELATED WORK

Modern, highly automated wafer fabs exhibit two levels of operational control. On the superordinate production process level, also named as machine level or fab level, a scheduler decides which wafer lots will be processed on which machine and when (Mönch et al. 2011). Between the process operations the wafer lots must be transported from one machine to another. This happens on the subordinate transport level, also named as AMHS level.

On the superordinate fab level, in many wafer fabs, the wafer lots are scheduled using dispatch methods (Scholl et al. 2011). For more efficient production schedules, shifting bottleneck heuristics (Mason et al. 2002), decomposition techniques (Ovacik and Uzsoy 1997) or metaheuristics can be applied (Sourirajan and Uzsoy 2007) to solve complex job shop scheduling problems (JSSP). In general, such fab level scheduling methods all seek to solve conflicts of wafer lots competing for scarce resources such as bottleneck machines. However, most such fab scheduling methods ignore the resource conflicts on the subordinate transport level. Overcoming this disadvantage, Drießel and Mönch (2012) optimized large-scale complex wafer lot scheduling problems consisting of process operations *and* transport operations using an extended disjunctive graph specialized for *vehicle-based* systems. Alternatively, Schmaler et al. (2017) simulate dispatch rules to forecast and schedule *vehicle-based* transports.

In contrast, we focus on *conveyor-based* systems as they may be considered for future 450 mm wafer production and as they are already successfully in operation in some 200 mm wafer fabs (Scholl et al. 2011). Nevertheless, few work exists on dispatching conveyor-based systems using production priorities (Wang et al. 2016). Alternatively, there exists work on simulating (Arzt and Bulcke 1999) or analytically estimating average material flows in conveyor-based AMHS (Nazzal et al. 2010). But most work on AMHS ignores the superordinate fab level. Instead, given traffic is assumed, i.e., arbitrarily set arrival rates of transport jobs. Overviews of AMHS technologies and layouts are provided by (Agrawal and Heragu 2006) and (Montoya-Torres 2006). Several comparative studies evaluate the benefits of conveyor-based AMHS in contrast to vehicle-based systems, e.g., (Brain et al. 1999; Temponi et al. 2012; Tung et al. 2013). These studies assume given average traffic as well.

Positioning this study, it can be concluded that most existing simulation work on conveyor-based AMHS assumes that wafer lots are served in a first-come-first-served (FCFS) manner when the wafer lots arrive at conveyor conjunctions such as rotary tables. But very few existing work regards the superordinate fab schedule when scheduling the transports within the AMHS. Hence, we introduced a transport scheduling problem and suggested two different novel concepts for scheduling the wafer lot transports, namely *TI* and *TO* (Schwenke and Kabitzsch 2017). The *TO* method exploits slacks from fab level, which indicate how much a process operation can be delayed without delaying the completion time of any (other) job in the fab schedule. For computing such slacks the critical path method (CPM) can be used. In this document we further investigate both methods using two new AMHS models that fit the fab model SET2, which was provided by the Measurement and Improvement of Manufacturing Capacity (MIMAC) project (Fowler and Robinson 1995). More importantly, we also investigate the resulting variability in transport-related delays because variability is a negative feature of wafer fabs.

In wafer fabs, variability is the quality of nonuniformity in the overall production (Hopp and Spearman 2011). For example, there is nonuniformity in inter arrival times of wafer lots and in process times as well as in transport times. Altogether, these portions of nonuniformity make individual cycle times of wafer lots variable and increase their average. Crucially, we solely focus on transport-related variability because transport scheduling methods can only influence or reduce transport-related delays. In general, variability needs to be combatted because it drives up average cycle times and inventory levels in a fab. The reason for this are propagating delays, which can never be recovered anymore. Moreover, a delayed wafer lot will drag this delay throughout its entire remaining process route and hold up many other subsequent wafer lots on downstream machines. Thus, one of the main objectives of this study is to quantify variability in transport times.

## 3 METHODS

### 3.1 Design of AMHS Models

The MIMAC project provided seven reference models of wafer fabs. These so-called MIMAC models resemble the typical settings and behavior of semiconductor manufacturing very well. Thus, the MIMAC models are widely used by researchers to simulate new dispatch rules or to investigate optimization problems of wafer fabs. Unfortunately, the MIMAC models do not contain AMHS elements. Instead, they focus on the fab scheduling level, and accordingly they provide, e.g., product mixes, release rates, wafer lot routes using machine groups and capacities of machine groups. Hence, in order to resemble transport-related waiting effects, an AMHS model has to be created. For this purpose we developed three AMHS models that are compatible with MIMAC model SET2.

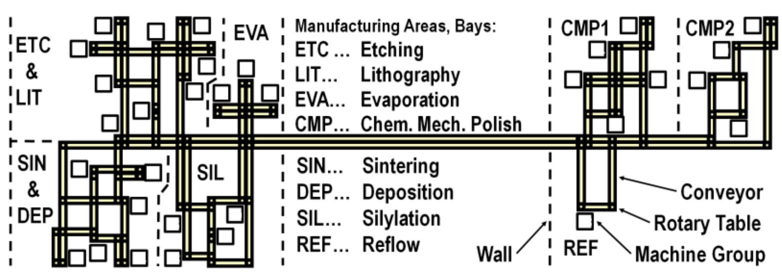


Figure 1: Specialized AMHS model A fitting transport times of SET2.

The first AMHS model (A) is described in (Schwenke and Kabitzsch 2017), see Figure 1. It is specialized for the use with SET2 because the locations of the machines are chosen in such a way that the AMHS resembles the given transport times of SET2. More precisely, this specialized AMHS model was created by mapping the given transfer times between production steps of process routes of SET2 to demanded distances between the corresponding machines. Subsequently, the locations of the machines were arranged in such a way that the majority of the machine pairs would resemble the demanded distance. In order to resolve contradictions, compromises needed to be found so that deviations were kept small. Additionally, if between two machines the travel time for going in one direction is different to the travel time for going in the opposite direction, two alternate one-tracked connections were placed between such machines. As a result, the layout of AMHS model A exhibits many shortcuts and bypasses. Thus, on the one hand it may be typical for mature wafer fabs, but on the other hand, at first sight, the layout appears interwoven and not quite straightforward. Hence, we designed two more generic layouts for a more general investigation of the suggested transport scheduling methods. Both new AMHS models are inspired by real fabs with spine layout. Thus, the new layouts are more generic because they resemble this common spine layout, also known as interbay-intrabay layout. Furthermore, more track elements are used because in reality the machines may not always be located for ideally short connections according to a given process route. For simplicity only a one-sided spine layout was created, which was big enough to house all machines of the example fab SET2. But the introduced transport scheduling methods could easily be used for scaled up models of bigger fabs. For a very much simplified baseline layout, the first new AMHS model (B) is only a simple loop for an interbay without any crossovers. In contrast, the second AMHS model (C) is more realistic because it exhibits crossovers in front of every bay. Furthermore, it contains bypasses in the back of the bays as it is the case in many conveyor-based AMHS.

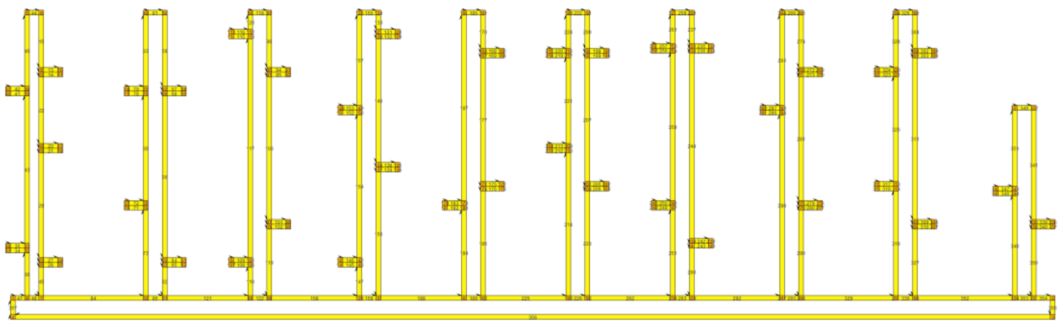

Figure 2: Generic AMHS model B without any crossovers or shortcuts.

The conveyors and rotary tables are 12 inch wide. Hence, for designing the AMHS models a drag-and-drop-based software tool was implemented so that productions areas (bays), machine groups as well as individual machines can be manually placed on a 12 inch grid. The models contain ten bays, which are 60 feet long and 24 feet wide. The connecting interbay double tracks are 220 feet long. All double tracks are 36 inches apart. The conveyor speed is set to 8.58 inches per second, whereas the transfer times on rotary tables are 6 seconds for passing straight and 9 in case of taking a turn.

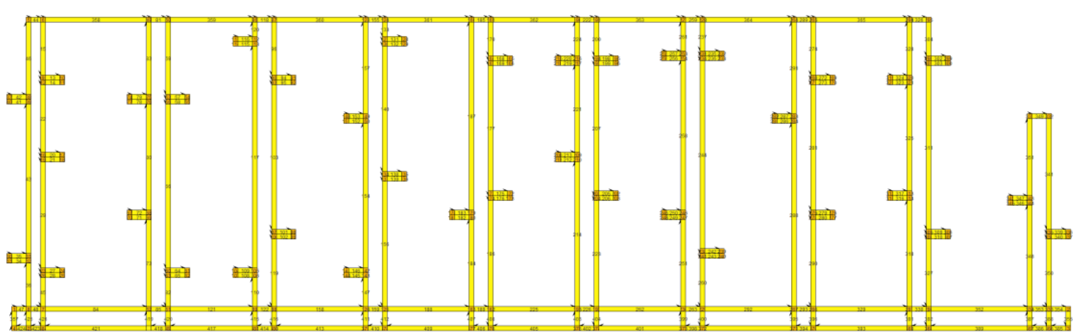

Figure 3: Generic AMHS model C with crossovers and bay shortcuts.

After bays and machines were manually placed, the connecting conveyors and rotary tables were placed automatically by the software tool. Additionally, all shortest paths between pairs of machines were computed and extracted. Subsequently, the intersecting parts of each two paths were computed for fast lookup of conflicts for later use in the transport scheduling methods. In Figures 2 and 3 the yellow long rectangles depict one-tracked conveyors and the squared orange rectangles symbolize rotary tables. For clarity, the machines are not displayed but they are located at the end of the horizontal load port branches at the vertical bays. Finally, the horizontal conveyors at the bottom of the pictures resemble the interbay.

### 3.2 Transport Insertion

Previously, we described the *TI* method only briefly (Schwenke and Kabitzsch 2017), hence, it is described in greater detail below. At first, symbols and indices are introduced: The overall fab schedule consists of production jobs $J_i$ with due dates $D_i$. A production job $J_i$ consists of process operations (steps) $O_{i\,o}$ so that index $i$ refers to the production job and the index $o$ refers to the step within $J_i$. Each operation $O_{i\,o}$ is associated with a starting time $s_{i\,o}$, a process duration $p_{i\,o}$ and a local due date $D_{i\,o}$. Between each two subsequent process operations $O_{i\,o}$ and $O_{i\,o+1}$, a transport job $J_k$ moves the wafer lot from the source machine to the destination machine. Hence, in contrast to the production jobs $J_i$ there are the transport jobs $J_k$, which consist of transport operations $T_{k\,t}$ so that index $k$ refers to the transport job and $t$ refers to the transport operation within a particular transport job. Each transport operation $T_{k\,t}$ is associated with an individual starting time $s_{k\,t}$ and duration $q_{k\,t}$ designating the entry of a rotary table or conveyor and the duration for traversing it. Thus, a transport operation $T_{k\,t}$ corresponds to the $t$-th traversing of a rotary table or conveyor on a wafer lot's path from one machine to another, and a transport $J_k$ consists of $n_k$ transport operations.

The *TI* method performs a basic loop, which successively constructs an overall transport schedule $S$ by inserting one transport job $J_k$ after another. In preparation to the loop, all transport jobs $J_k$ are sorted in non-decreasing order of their ready times $r_k$ and then they are stored in the set $U$ of all unscheduled transports. A ready time $r_k$ is the point in time, when a transport can start, which is not earlier than the finish of its preceding process operation $O_{i\,o}$. Accordingly, when the transport job $J_k$ arrives at its destination machine, a subsequent process operation $O_{i\,o+1}$ will be executed. Subsequently, in each iteration of the loop, the next unscheduled transport job is inserted into the next earliest possible time period that allows unimpeded movement of its wafer lot across all conveyors and rotary tables on its travel path from its source machine to its destination machine.

When a transport $J_k$ is inserted into the transport schedule $S$, three things happen. Firstly, all transport operations $T_{k\,t}$ of $J_k$ are inserted at once. Most importantly, the transport operations $T_{k\,t}$ are inserted without any delay in between them, i.e., $s_{k\,t} + q_{kt} = s_{k\,t+1}$, which is the no-wait constraint between the transport operations. This no-wait constraint enforces the wafer lots to travel unimpeded so that congestions are avoided. Hence, possibly a waiting time $w_k$ may need to be inserted before the transport $J_k$ actually starts moving, i.e., $r_k + w_k = s_{k1}$. As a result, all succeeding transport operations $T_{k\,t}$ of transport job $J_k$ are fixed as well.

Secondly, after the transport is fixed the delay $w_k$ must be checked if it causes the transport to arrive too late at its destination machine. It arrives too late if the local due date $D_{i\,o+1}$ of the succeeding process operation $O_{i\,o+1}$ cannot be met anymore. Note, in preparation of the loop, the local due date $D_{i\,o}$ was computed beforehand for each operation $O_{i\,o}$ of the overall fab schedule by applying the critical path method. More specifically, a transport arrives too late if its arrival time $a_k = s_{k\,n_k} + q_{k\,n_k}$ causes the operation $O_{i\,o+1}$ to end after its local due date $D_{i\,o+1}$, i.e., $a_k > D_{i\,o+1} - p_{i\,o+1}$. For convenience, the local due date $D_{i\,o+1}$ can be converted into the arrival due date $d_k$ of the transport $J_k$, which is the point in time when the transport should arrive at the latest, i.e., $d_k = D_{i\,o+1} - p_{i\,o+1}$. If the transport arrives too late, then the starting time $s_{i\,o+1}$ of process operation $O_{i\,o+1}$ is not valid anymore. Hence, it must be corrected to $s'_{i\,o+1} := r_k + w_k + \sum_{t=0}^{n_k} q_{kt}$. Note, $\sum_{t=0}^{n_k} q_{kt}$ is the raw transport time of $J_k$. Furthermore, all subsequent process operations' starting times of the overall fab schedule need to be updated, using the CPM, as well because the delay might propagate. After updating the fab schedule, the next transport in the sorted list might start delayed as well, but the order of transport starts will be maintained.

Thirdly, for each rotary table it needs to be stored, when it is occupied, or 'booked'. For this purpose, for each rotary table $Q$, there exist two ordered sets. The first set $(E_Q, \leq)$ in non-decreasing order contains all points in time $e$ when wafer lots enter. The second ordered set $(F_Q, \leq)$ contains the corresponding exit times $f$. Hence, together the two sets keep the information when a rotary table is occupied. A transport operation $T_{k\,t}$ has to fit between the end $f$ of another previous transport operation of another transport job and the beginning $e$ of a subsequent transport operation of another transport job. The *TI* method is noted in pseudocode below.

1. Sort transports $J_k$ by ready times $r_k$ and store them in set $U$ of unscheduled transport jobs.
2. Pick first transport job $J_k$ from set $U$, i.e., $U := U \setminus J_k$.
3. Find time periods for $J_k$ in transport schedule $S$, so that for each transport operation $T_{k\,t}$ holds:
   (a) $s_{k\,t} > f \in F_Q$ and
   (b) $s_{k\,t} + q_{k\,t} < e \in E_Q$
4. For each transport operation $T_{k\,t}$ add the found starting times $s_{k\,t}$ and end times $s_{k\,t} + q_{k\,t}$ to the ordered sets of starting times $E_Q$ and end times $F_Q$ of occupation of the associated rotary table $Q$.
5. Fix the starting times $s_{k\,t}$ of transport $J_k$ in schedule $S$.
6. If the arrival $a_k = r_k + w_k + \sum_{t=0}^{n_k} q_{kt}$ of $J_k$ is later than the start $s_{i\,o+1}$ of the subsequent process operation $O_{i\,o+1}$, i.e., $a_k > s_{i\,o+1}$, then update the fab schedule using the CPM.
7. If $U \neq \{0\}$ then goto 2 else stop.

All transport jobs are sorted and stored in the ordered set $U$ (1). If a transport shall be inserted into the overall transport schedule $S$, it is deleted from $U$ (2). Accordingly, a transport can only occupy a rotary table if the preceding transport left, i.e., $s_{k\,t} > f \in F_Q$, and a possibly already allocated succeeding transport has not entered yet, i.e., $s_{k\,t} + q_{k\,t} < e \in E_Q$. Thus, periods in time have to be found when each succeeding rotary table $Q$ of a transport $J_k$ is unoccupied (3). Crucially, such periods of time have to be subsequent without any delay across several rotary tables so that they obey the no-wait restriction which demands the unimpeded moving of transport job $J_k$. Once such a succession of time periods is found, the rotary tables need to be booked (4). Afterwards, the starting times of the current transport job $J_k$ are fixed (5). At last, it must be checked if the succeeding process operation can be started as scheduled or if the transport arrives delayed. If so, the fab schedule must be updated (6). If all transport jobs have been scheduled (inserted into schedule $S$), the algorithm stops (7). In summary, during the execution of the *TI* method the transport schedule $S$ will be filled sequentially.

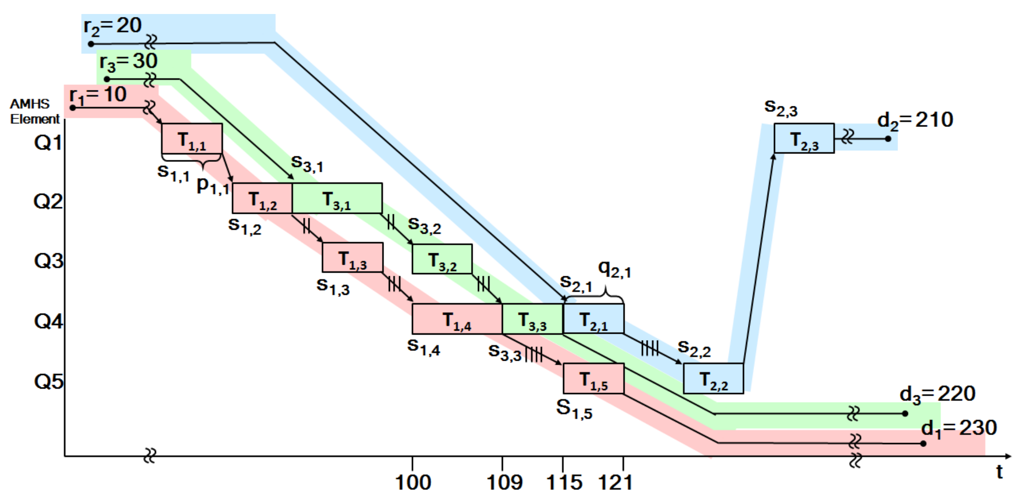


Figure 4: The transport schedule $S$ is filled successively with three transports $J_1$, $J_2$ and $J_3$. The order in which the transports are inserted is according to their ready times $r_1$, $r_2$ and $r_3$.

Figure 4 shows an example of three transport jobs being inserted one after another into an overall transport schedule. The schedule is depicted in the form of a Gantt chart. There are five rotary tables involved, i.e., $Q_1$ to $Q_5$, where the transport jobs have to cross, see Figure 5. Hence, the schedule in Figure 4 is only a small extract of the complete overall transport schedule. It only covers the events on the rotary tables during the time period when these three transport jobs traverse. The first job is $J_1$ because it is ready to start at ready time $r_1 = 10$. Thus, it is inserted undelayed and traverses the five rotary tables at times $s_{1,1}$ to $s_{1,5}$. The second transport job $J_2$ starts at ready time $r_2 = 20$ and traverses a long distance of conveyor belts before it arrives at its rotary tables $Q_4$ at point in time 115 seconds. Then $J_2$ traverses $Q_4$ *and* $Q_5$ unimpeded as well. The last job $J_3$ is ready at $r_3 = 30$ and must find sufficient time windows on $Q_2$, $Q_3$ and $Q_4$. It can travel across $Q_2$ only after $J_1$ left. It reaches $Q_3$ clearly after $J_2$ left. At $Q_4$ it has to be checked if the preceding job $J_1$ left early enough. Moreover, it has to be checked if the succeeding job $J_2$ arrives late enough. Importantly, only if this is the case, it can fit between $J_1$ and $J_2$ on $Q_4$. For instance, if $J_2$ would arrive earlier, then job $J_3$ could not enter $Q_4$ immediately. Instead it would have to introduce more waiting time $w_3$ so that $J_3$ appears at $Q_4$ only after $J_2$ left. In this case, $J_3$ would experience a much larger delay.

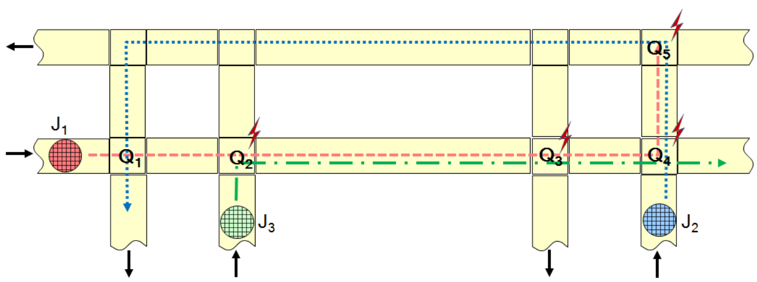


Figure 5: Extract of interbay and conjunct bays with conveyors (rectangles) and rotary tables (squares). Three transport jobs $J_1$, $J_2$ and $J_3$ traverse rotary tables $Q_1$ to $Q_5$ and form conflicts (lightning bolts).

### 3.3 Transport Optimization

In contrast to the *TI* method, the *TO* method always considers several transports at once. More specifically, it jointly considers such transports that constitute conflicts. Hence, the *TO* method solves a JSSP optimally by minimizing the total weighted completion time of the transport jobs so that the most urgent transport jobs are weighted the most and thus, will be delayed the least. The *TO* method is comprehensively described in (Schwenke and Kabitzsch 2017), in which it is called MIP solving method. Therefore, below only the conceptual differences to the *TI* method are highlighted focusing on finding conflicts and stating a corresponding JSSP.

The conflicts are discovered as follows. Initially, in preparation of a basic loop, the transports are sorted in non-decreasing order by their ready times $r_k$ in the same manner as at the beginning of the *TI* method. Then all transports that leave after marked process operations are checked with each other to determine if they constitute conflicts. A process operation will be marked if its preceding transport has been scheduled by the method or if it has no preceding transport such as the first operation $O_{i\,1}$ in a production job $J_i$.

Transports constitute conflicts if the durations of transport operations on rotary tables would overlap. For example, if transport jobs $J_k$ and $J_l$ start both directly after their preceding process operations at their ready times $r_k$ and $r_l$, then their transport operations $T_{k\,t}$ and $T_{l\,u}$ overlap if there exist any step $t$ in transport $J_k$ and step $u$ in transport $J_l$ so that $(s_{l\,u} \geq s_{k\,t}$ *AND* $s_{l\,u} < s_{k\,t}+q_{k\,t})$ *OR* $(s_{k\,t} \geq s_{lu}$ *AND* $s_{k\,t} < s_{l\,u} + q_{l\,u})$. For example, in Figure 5 three transport jobs $J_1$, $J_2$ and $J_3$ would simultaneously try to traverse the rotary table $Q_4$, because each of them would arrive at $Q_4$ at the same point in time, e.g., at 100 seconds. Furthermore, the transports form conflicts at rotary tables $Q_2$, $Q_3$ and $Q_5$. Note, $J_1$ and $J_2$ do not constitute a conflict at $Q_1$ because their time windows of their transport operations $T_{1,1}$ and $T_{2,3}$ for traversing $Q_1$ do not overlap. In summary, the three transports compete for resources so that they constitute a JSSP.

Therefore, the algorithm collects all transports that form conflicts with each other in a problem set $P$ of transports to be simultaneously scheduled. Then for this subset $P$ of transports a JSSP is stated and formulated as a mixed integer problem (MIP). Subsequently, the JSSP is solved by a standard MIP solver. In order to avoid congestions, the transports shall travel unimpeded, i.e., the JSSP is subject to no-wait constraints. In order to advantage urgent transports the local due dates of the subsequent process operations of each transport are regarded in the target function of the MIP. As mentioned, for convenience, this local process operation due date previously is converted into the due date of the transport $d_k$, which is the point in time before the transport should arrive. The *TO* method is noted in pseudocode as follows.

1. Sort transports $J_k$ by ready times $r_k$ and store them in set $U$ of unscheduled transport jobs.
2. Collect transports $J_k$ that constitute conflicts:
   (a) Consider first transport $J_k$ from set $U$.
   (b) For each transport $J_l \in P$ check if $J_l$ and $J_k$ constitute conflicts, i.e.,
   if $(s_{l\,u} \geq s_{k\,t}$ *AND* $s_{l\,u} < s_{k\,t}+q_{k\,t})$ *OR* $(s_{k\,t} \geq s_{lu}$ *AND* $s_{k\,t} < s_{l\,u} + q_{l\,u})$
   then Add $J_k$ to set $P$ of transports to be scheduled, i.e., $P = P \cup \{J_k\}$ and
   Remove $J_k$ from set $U$, i.e., $U := U \backslash J_k$ and
   goto 2(a)
   else goto 3.
3. For $P$ state a JSSP with no-wait constraints and derive a MIP formulation.
4. Solve the MIP using a MIP solver.
5. For each $J_k$ in the solution of the JSSP: Fix the starting times of transport $J_k$ in schedule $S$.
6. For each $J_k$ in the solution of the JSSP check if arrival $a_k = r_k + w_k + \sum_{t=0}^{n_k} q_{kt}$ of $J_k$ is later than the start $s_{i\,o+1}$ of the subsequent process operation $O_{i\,o+1}$, i.e.,
   if $a_k > s_{i\,o+1}$ then update the fab schedule using the CPM.
7. If $U \neq \{0\}$ then goto 2 else stop.

In short, the solution of the JSSP is a priority order for conflicting transports to be scheduled. Thus, at step 5 the pseudocode is very similar to the *TI* method. The only difference is that the ordered sets $(E_Q, \preceq)$ and $(F_Q, \preceq)$ for checking when the rotary tables are booked are not needed because the MIP solution of the JSSP already guarantees conflict free movement. The set of transports that form conflicts is rather small, e.g., 20 transports, compared to all transports to be scheduled in a large-scale fab scheduling problem. Hence, a rather small size of JSSPs is a prerequisite of the suggested approach. Otherwise, computing an exact solution using a MIP solver would take too long. As a fallback heuristic in case of an intractable MIP, the *TI* method could be used or the common FCFS policy at the rotary tables.

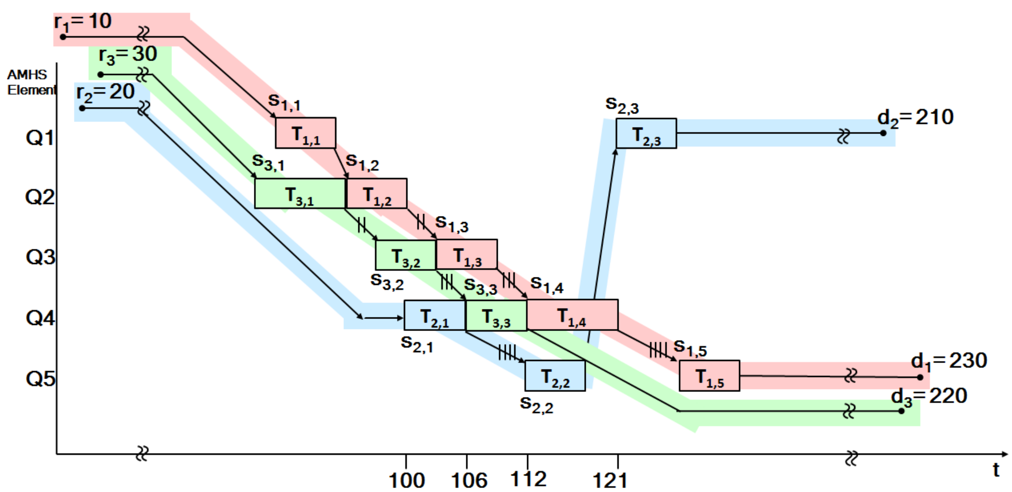

Figure 6: Transport schedule of three transports $J_1$, $J_2$ and $J_3$ as it is yielded by solving a JSSP.

For example, Figure 6 depicts the solution of a very small MIP. If the transports $J_1$, $J_2$ and $J_3$ would be inserted into the schedule $S$ in sequence of their ready times $r_1$, $r_2$ and $r_3$ they would form a different solution schedule as it is depicted in Figure 4 for the *TI* method. Instead, the JSSP takes the due dates of the transports into account and computes the best compromise. As a result, e.g., the transports traverse rotary table $Q_4$ in a different sequence. Accordingly, the arrival times of the transports at their destination machines are different. For instance, the most urgent transport $J_2$ arrives much earlier and can meet its due date $d_2$=210 much better. In summary, the conceptual differences between the *TI* method and the *TO* method are as follows. Firstly, the *TI* method only handles one transport at once. Conflicts between transports are simply solved by delaying the subsequent starting transport job. Hence, the method is uninformed of other transport jobs, which is a disadvantage. In contrast, the *TO* method always handles several conflicting transports at once. Thus, it finds a compromise between the competing transports, which is clearly an advantage.

Secondly, the computational cost of solving the MIPs can be a disadvantage but is bearable in most cases. Therefore, the *TI* method exhibits a small advantage in regard to computation time. Thirdly, the *TI* method must keep lists for logging when the rotary tables are occupied by a wafer lot so that their limited capacity is regarded. In contrast, the *TO* method does not need such lists, because it intrinsically regards the limited capacity of rotary tables by the classic capacity constraints within the JSSP.

Similarities are that both methods schedule the transport jobs in a loop, which processes a previously produced fab schedule. Both methods regard no-wait constraints so that the transports remain unimpeded after they started. Consequently, both methods introduce waiting times after the ready times of the transport jobs and before the transport jobs actually start.

## 4 VALIDATION

For validation of both transport scheduling methods, three performance indicators were determined, namely transport-related delay of cycle times, total number of delayed transports and variability. In order to produce input data, the MIMAC model SET2 was used to simulate dispatch rules. Importantly, the dispatch rules consider the idealized raw transport times according to the applied AMHS. A raw transport time is the time a wafer lot would take to travel from one machine to another if there were no other wafer lots in the way.

Then the produced fab schedule was used to induce transport jobs to transfer a wafer lot after each process operation to the machine of its next process operation. Hence, for simplicity idle transports are ignored. Fab schedules were produced by simulation of the dispatch rules shortest processing time (SPT), first come first served (FCFS) and earliest due date (EDD) so that results are averaged. The scheduling horizon of the tests is 5 days using product mix and release rates as given in the MIMAC data set. As a result, 31 jobs with an average of 250 operations using 277 machines, grouped in 97 sets, were produced. A corresponding fab level schedule with a total of 7656 process operations was constructed inducing 7625 transport jobs. Furthermore, three different AMHS models, as described in Subsection 3.1, were applied for conducting the experiments. Table 1 shows the applied AMHS (A, B or C) as well as the investigated transport scheduling method of the conducted experiments.

Table 1: Matrix of conducted experiments.

| transport scheduling method | applied AMHS model | | |
|---|---|---|---|
| | A | B | C |
| First Come First Serve | X | | |
| Transport Insertion | X | X | X |
| Transport Optimization | X | X | X |

Most importantly, after a fab schedule is produced, the order of the process operation within this fab schedule is fixed, i.e., the order is never changed anymore by the transport scheduling methods. Instead, the transport scheduling methods only introduce transport-related delays which originate from solving the conflicts as described in Subsections 3.2 and 3.3. More precisely, it is analyzed how much a particular transport scheduling method increases the cycle time of a process job compared to the idealized, but unrealistic, case of raw transport times. This case is unrealistic because it would be assumed that wafer lots never block each other and transport conflicts would never form.

The first set of experiments was carried out similarly to the validation in (Schwenke and Kabitzsch 2017). Additionally, now the variability of the transport-related delay is investigated. Hopp and Spearman (2011) suggest to use variation coefficients $c$ to quantify variability. Hence, they define three degrees. Low variability corresponds to $c < 0.75$. Medium variability is $0.75 \leq c < 1.33$. High variability corresponds to

$c \geq 1.33$. Thus, Table 2 shows for each method the reduction of the transport-related delays in a process job's cycle time. Furthermore, it displays the number of delayed transports, the total transport-related delays in seconds as well as the average delay $d$ in seconds. Accordingly, the variation coefficient can be derived as the normalized standard deviation, i.e., $c = \sigma/d$. As a result, Table 2 shows that the variability of transport-related delays is significantly reduced by applying the transport scheduling methods *TI* and *TO* in comparison to the baseline method FCFS. Hence, a conveyor-based system that is operated by use of FCFS rules at rotary tables already exhibits low variability. But the methods *TI* and *TO* can lower the variability even more.

Table 2: Variability in transport-related delays using AMHS model A.

| | ***FCFS (baseline)*** | ***Transport Insertion*** | ***Transport Optimization*** |
|---|---|---|---|
| Reduction of transport-related delays vs. baseline method (FCFS) in % | 0% | 64% | 82% |
| Number of delayed transports | 251 | 122 | 95 |
| Total transport-related delays in seconds | 2109 | 810 | 416 |
| Average transport-related delay $d$ in seconds | 8.40 | 6.63 | 4.73 |
| Standard deviation of transport-related delays $\sigma$ | 6.03 | 4.31 | 2.40 |
| Transport-related variability $c = \sigma/d$ | **0.72** | **0.65** | **0.55** |

The second set of experiments particularly focused on the comparison of the two novel methods *TI* and *TO*. For this purpose, the remaining validation uses the new AMHS layouts B and C, see Table 3. The experiments, show that the average transport-related delays $d$ can be driven down more by the *TO* method, even though there are more delayed transports if the *TO* method is applied. Hence, the extent of the delay in a delayed transport is much smaller if the *TO* method is used. Similarly, the standard deviation $\sigma$ of transport-related delays is smaller for the *TO* method ($\sigma = 3.37$) than for *TI* ($\sigma = 5.44$) if AMHS model C is used.

Table 3: Variability in transport-related delays using AMHS models B and C.

| | ***Transport Insertion*** | | ***Transport Optimization*** | |
|---|---|---|---|---|
| AMHS model | B | C | B | C |
| Number of delayed transports | 265 | 178 | 321 | 193 |
| Total transport-related delays in seconds | 2387 | 1426 | 1972 | 977 |
| Average transport-related delay $d$ in seconds | 9.01 | 8.01 | 6.14 | 5.06 |
| Standard deviation of transport-related delays $\sigma$ | 5.16 | 5.44 | 5.05 | 3.37 |
| Transport-related variability $c = \sigma/d$ | **0.57** | **0.67** | **0.82** | **0.66** |

Furthermore, Table 3 shows for *TO* that AMHS model C leads to better results than AMHS model B, which was expected because more shortcuts and crossovers lead to more distinct paths and thus to less occasions for conflicts to be solved. Surprisingly, in terms of variability $c$ the results do not show as clearly. The reason is the normalization in the formula for the variation coefficient $c = \sigma/d$. For example, using AMHS model B, even though absolute values of delay $d$ and standard deviation $\sigma$ are smaller for *TO* ($d = 6.14$, $\sigma = 5.05$) than for *TI* ($d = 9.01$, $\sigma = 5.16$) the normalized values of $c$ indicate higher variability for *TO* ($c = 0.82$) than for *TI* ($c = 0.57$). Hence, normalization can skew the picture. Similarly, using AMHS model C, *TO* leads to similar results ($c = 0.66$) as *TI* ($c = 0.67$), but the absolute value of standard deviation is much better for *TO* ($\sigma = 3.37$) than for *TI* ($\sigma = 5.44$).

## 5 CONCLUSIONS

Two transport scheduling methods for fixing the transport operations between the process operations of a wafer fab are investigated and compared. The first scheduling method, called *TI*, sequentially constructs a transport schedule by inserting one transport after another into a comprehensive transport schedule. In contrast, the second method is called *TO* because it simultaneously optimizes and fixes several conflicting transports so that the most urgent transport jobs will be advantaged. Both methods derive a priority order of transports to be scheduled. Hence *TI* can be seen as a special case of *TO*. The inner works of both methods are explained using an illustrative example.

Validation comprises computational experiments for both transport scheduling methods. Besides total and average transport-related delays, the variability in such transport-related delays is examined because in wafer fabs the pursuit of lower variability is a continuous battle. For the experiments, three different AMHS models are used. All AMHS models exhibit the common interbay-intrabay layout, but the first AMHS model is specialized for the use with only one MIMAC model. Therefore, two more generic AMHS models are designed. In summary, the computational experiments, show that average transport-related delays $d$ and standard deviation $\sigma$ can be driven down more by *TO* than by *TI*. Similarly, using an AMHS model with more shortcuts and crossovers leads to better results as well. In terms of variability the differences do not show as clearly because of the normalization in its formula $c = \sigma/d$. Hence, normalization can obfuscate the results, but in general both transport scheduling methods lead to low or to lower medium variability.

The limitations of this study are threefold. Firstly, the quantification of variability may strongly depend on the structure of the AMHS. Secondly, no idle transports are considered. Thirdly, the scale of the AMHS and fab model may be smaller than in current mass producing wafer fabs. Nevertheless, both methods hold the potential to be adjusted accordingly. Furthermore, both methods exhibit the advantage that they suppress congestions by enforcing unimpeded transports by application of a no-wait constraint between a transport job's subsequent transport operations. As a result, the transport jobs may start delayed but then travel unimpeded. Hence, congestion free transportation can be achieved by buying a small waiting time before beginning the transport jobs.

## ACKNOWLEDGMENTS

This work was partly funded via the research projects SemI40 agreement# 16ESE0077S (ECSEL Joint Undertaking, EU Horizon 2020, BMBF,SMWA) and IWAHAF agreement# and 01IS16035B (BMBF).

## REFERENCES

Agrawal, G. K. and S. S. Heragu. 2006. "A Survey of Automated Material Handling Systems in 300-mm Semiconductor Fabs". *IEEE Transactions on Semiconductor Manufacturing*, 19(1):112–120.

Arzt, T. and F. Bulcke. 1999. "A New Low Cost Approach in 200 mm and 300 mm AMHS". *Semiconductor Fabtech* 10:19–26.

Drießel, R., L. and Mönch. 2012. "An Integrated Scheduling and Material-Handling Approach for Complex Job Shops: a Computational Study". *International Journal of Production Research* 50(20):5966–5985.

Hopp, W. J. and M. L. Spearman. 2011. *Factory Physics*. Illinois, US: Waveland Press, Inc.

Mason, S. J., J. W. Fowler, and W. M. Carlyle. 2002. "A Modified Shifting Bottleneck Heuristic for Minimizing the Total Weighted Tardiness in a Semiconductor Wafer Fab". *Journal of Scheduling* 5(3):247–262.

Mönch, L., J. W. Fowler, S. Dauzere-Peres, S. J. Mason, and O. Rose. 2011. "A Survey of Problems, Solution Techniques, and Future Challenges in Scheduling Semiconductor Manufacturing Operations". *Journal of Scheduling* 14(6):583–599.

Montoya-Torres, J. 2006. "A Literature Survey on the Design Approaches and Operational Issues of Automated Wafer-Transport Systems for Wafer Fabs". *Production Planning and Control* 17(7):648–663.

Nazzal, D., J. Jimenez, H. Carlo, A. Johnson, and V. Lasrado. 2010. "An Analytical Model for Conveyor-Based Material Handling System with Crossovers in Semiconductor Wafer Fabs". *IEEE Transactions on Semiconductor Manufacturing* 23(3):468–476.

Ovacik, I. and Uzsoy, R. 1997. *Decomposition Methods for Complex Factory Scheduling Problems*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Schmaler, R., T. Schmidt, M. Schoeps, J. Luebke, R. Hupfer, R. and N. Schlaus. 2017. "Simulation based evaluation of different empty vehicle management strategies with considering future transport jobs". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan et al., 3576-3587. Piscataway, NJ: IEEE.

Scholl, W., B.-P. Gan, P. Lendermann, D. Noack, O. Rose, P. Preuss, and F. Pappert. 2011. "Implementation of a Simulation-based Short-term Lot Arrival Forecast in a Mature 200mm Semiconductor Fab". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain et al., 1927–1938. Piscataway, NJ: IEEE.

Schwenke, C. and Kabitzsch, K. "Continuous Flow Transport Scheduling for Conveyor-Based AMHS in Wafer Fabs". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan et al., 3588–3599. Piscataway, NJ: IEEE.

Sourirajan, K. and R. Uzsoy. 2007. "Hybrid Decomposition Heuristics for Solving Large-Scale Scheduling Problems in Semiconductor Wafer Fabrication". *Journal of Scheduling* 10(1):41–65.

Temponi, C., J. A. Jimenez, and F. A. M. Mediavilla. 2012. "Critical Variables in the Decision-Making Process for AMHS Technology Selection in Semiconductor Wafer Size Transitions: Exploratory Study". *IEEE Transactions on Semiconductor Manufacturing* 25(3):408–419.

Tung, J., Sheen, T., Kao, M. and Chen, C. 2013. "Optimization of AMHS Design for a Semiconductor Foundry Fab by Using Simulation Modeling". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy et al., 3829–3839. Piscataway, NJ: IEEE.

Wang, C.-N., Y.-T. Chung, Y.-H. Wang, M.-T. Duong, and T.-F. Lin. 2016. "The Material Dispatching Method for Conveyor System in 450 mm Wafer Fabrication". *Journal of Testing and Evaluation* 45(3):940-952.

## AUTHOR BIOGRAPHIES

**CLEMENS SCHWENKE** received his M.S. in Electrical Engineering from Dresden University of Technology. He is a Ph.D. student at the Chair of Technical Information Systems of Professor Klaus Kabitzsch. His research interests include modeling, simulation and scheduling in automation. His e-mail is clemens.schwenke@tu-dresden.de.

**SEBASTIAN JANNASCH** received his B.S. in Computer Science from Dresden University of Technology. His research interests include simulation of AMHS. His e-mail is sebastian.jannasch@tu-dresden.de.

**KLAUS KABITZSCH** holds the Chair of Technical Information Systems at the Institute of Applied Computer Science of the Dresden University of Technology, Germany. He received a Diploma and a Ph.D. in Electrical Engineering and Communications Technology. His current projects focus on software tools for design of networked automation, data analysis, advanced process control and predictive technologies. He is a member of IEEE, VDE and GI. His e-mail is klaus.kabitzsch@tu-dresden.de.