Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

SIMULATION-BASED OPTIMIZATION APPROACH FOR SOLVING PRODUCTION PLANNING PROBLEMS

Hajime Sekiya,[1] and Lars Mönch[1]

[1]Dept. of Mathematics and Computer Science, University of Hagen, Hagen, GERMANY

ABSTRACT

Production planning for wafer fabs often relies on linear programming. Exogenous lead time estimates or workload-dependent lead times by means of clearing functions are taken into account in common planning formulations. For realistic performance assessment purposes, the process uncertainty is captured by simulating the execution of the resulting release schedules, i.e. expected values for profit or costs are considered. In the present paper, we take a more direct approach using simulation-based optimization. The capacity constraints and the lead time representation are indirectly respected by executing release schedules in a simulation model of a large-scaled wafer fab. Variable neighborhood search (VNS) is used to compute release schedules. We show by designed experiments that the proposed approach is able to outperform the allocated clearing function (ACF) formulation for production planning under many experimental conditions.

1 INTRODUCTION

Production planning is an important planning function in semiconductor supply chains. It deals with deciding which amount of a certain product should be released into a single semiconductor wafer fabrication facility (wafer fab) over time to optimize some performance measure of interest, such as cost or profit to meet demand based on output from master planning (Mönch et al. 2018).

Production planning formulations are often based on deterministic data for capacity and process flows derived from data found in manufacturing execution systems. Since cycle times (CTs), the time span between releasing work and its emergence as final product, is of the order of ten weeks in most wafer fabs, lead time (LT) information as estimates of the CT must be incorporated into production planning formulations for wafer fabs (Leachman 2000). Exogenous LTs are distinguished from workload-dependent LTs. The CTs are determined by the release decisions of the planning formulation. Therefore, LT information is an output of the planning formulation rather than an input parameter. Planning formulations based on exogenous LTs are often outperformed by planning models with workload-dependent LTs when the production plans are executed in a stochastic environment (Kacar et al. 2013; Kacar et al. 2016).

Nonlinear clearing functions (CFs) model the expected throughput of a production resource as a function of its planned workload. CF-based planning models have yielded promising results when used in production planning models if the CFs are correctly parameterized. The resulting formulations after piecewise linearization of the CF-related constraints are large-scaled linear programs (LPs) which can be solved efficiently by commercial solver software. Recent approaches apply also non-linear optimization techniques such as conic programming and avoid a linearization of the original non-linear formulations (cf., for instance, Gopalswamy and Uzsoy 2020).

The situation changes drastically when building a deterministic planning model from shop-floor and environmental data is difficult or when integer-valued decision variables have to be included into planning models. These modeling and solution difficulties can be avoided to some extent when discrete-event simulation is used to represent the base system of a wafer fab and its environment within a simulation-based optimization approach. While it is likely that such an approach will work in principle, it is not clear whether

the computational burden of simulating large-scale wafer fabs is a barrier for the simulation-based optimization approach or not. Therefore, in the present paper we will design and test a simulation-based optimization approach for a stylized planning problem for a large-sized wafer fab.

The paper is organized as follows. In the next section, we describe the production planning problem at hand, discuss related work, and derive research questions. The different production planning approaches, namely the ACF formulation and the VNS scheme, are discussed in Section 3. Implementation aspects are discussed in Section 4. The results of computational experiments are reported in Section 5. Finally, conclusions and future research directions are provided in Section 6.

2 PROBLEM SETTING AND ANALYSIS

2.1 Production Planning Problem

Production planning involves the allocation of available capacity among the operations of the products to match supply with given demand in some near-optimal manner. Release decisions are made for a wafer fab. However, it is known from queuing theory, discrete-event simulation, and industrial observations that the mean and the variance of the CT increase nonlinearly with resource utilization, which, in turn, is determined by the release decisions made by the production planning function. This circularity implies that CTs are an output of production planning rather than an input. Hence, CTs are variables to be controlled in planning models, rather than exogenous parameters that must be estimated.

Conventional production planning models described in the literature are based on exogenous LTs, fixed parameters independent of the congestion of the wafer fab. This approach leads to computationally tractable LP models, but fails to represent the congestion of the wafer fab correctly. There is research that explicitly addresses this circularity by modeling workload-dependent LTs in production planning models. Iterative methods combine LP models with exogenous LTs with simulation, queuing, or scheduling models to update LTs (Missbauer and Uzsoy 2020). But the convergence behavior of these methods is unclear (Missbauer 2020). Nonlinear optimization models based on queueing concepts to represent the cost of congestion form another class of approaches. Among them CF-based models are popular. A CF estimates the average output of a work center in a planning period as a function of its available workload in that period. While early CF-based models had difficulties to deal with multiple products, the ACF formulation by Asmundsson et al. (2009) addresses this situation. One limitation of CF-based production planning approaches is that yet no rigorous methodology for estimating CFs from data is known. This can be seen as a large barrier to their widespread adoption in planning models (Gopalswamy and Uzsoy 2019). Another class of production planning models are data-driven (DD) formulations (cf., for instance, Völker and Mönch 2023) which are based on a set of system states representing the congestion behavior of a wafer fab with work in progress (WIP) and resulting output levels. DD formulations can be seen as an alternative to CF-based production planning formulations (Missbauer and Uzsoy 2020). Large-sized mixed-integer linear programming (MILP) formulations are the result of the need to choose system states.

However, the approaches discussed so far implicitly assume that a deterministic model can be built that anticipates the behavior of the base system, i.e. the shop floor, and the external environment of a wafer fab, correctly. This assumption is not always valid. Another problem are integer-valued decision variables in planning formulation which make the formulations computationally intractable. For instance, Ziarnetzky et al. (2017) propose a production planning model for wafer fabs where the installation of renewable energy sources such as wind turbines (WTs) or photovoltaic panels (PVs) is considered. Both the energy consumption of the machines on the shop floor and the energy provided by wind and sun are difficult to model without simulation. Moreover, the amount of WTs and PVs must be modeled by integer-valued decision variables. Werner et al. (2022) consider a related long-term strategic planning model for semiconductor supply chains. Again, large-scaled MILP models must be solved. One possible approach to deal with these problems is simulation-based optimization where the wafer fabs and their environment are represented by a simulation model which represents important constraints and the process uncertainty from the shop floor and the environment.

2.2 Related Work

Next, we discuss known simulation-based optimization approaches for wafer fabs. Liu et al. (2011) use simulation-based optimization based on the Nondominated Sorting Genetic Algorithm II (NSGA-II) to determine Pareto-optimal production plans for the expected value and variance of the sum of WIP, inventory holding, and backlog cost. A simulation model of a scaled-down wafer fab which contains only 11 machines is used. Ziarnetzky and Mönch (2016) consider a simplified semiconductor supply chain consisting of a wafer fab and a backend facility. Simulation-based optimization based on simulated annealing is used to determine the minimum bottleneck utilization in the wafer fab and the amount of expanded capacity in the BE facility. A simulation model of a large-sized wafer fab with more than 200 machines is used for this design problem. Kacar and Uzsoy (2015) use a gradient-based algorithm to determine appropriate parameters for the CFs in the ACF formulation by simulation-based optimization. Moreover, the same technique is used for determining release plans. The production plans obtained by simulation-based optimization outperforms the ones obtained by the CF-based methods including the one where the CF are obtained from simulation-based optimization. Zhang et al. (2022) apply simulation-based optimization to improve CFs in a production planning model. In both papers, the proposed methods are applied to a simulation model of a scaled-down wafer fab which contains only 11 machines. Overall, we conclude that most of the simulation-based optimization schemes for production planning of wafer fabs are based only on scaled-down manufacturing systems. Therefore, in the present paper, we propose a simulation-based optimization approach for a simulation-model of a large-sized wafer fab.

3 SOLUTION APPROACHES

3.1 ACF Formulation

We assume a planning window of T periods of the same length. For the sake of completeness, we repeat the ACF model. It is used for benchmarking the simulation-based optimization scheme. Linearized CFs constrain the achievable output quantity for each work center k in units of time, allocating it to the products g and operations $l \in O(g,k)$. The formulation is based on the following sets and indices, decision variables, and parameters.

Sets and indices:

t:	period index with planning window T
g:	product index
G:	set of all products g
k:	work center index
K:	set of all work centers
l:	operation index
$O(g)$:	set of all operations of product g
$O(g,k)$:	set of all operations of product g that can be performed on machines of work center k
$K(g,l)$:	set of work centers that can be used to perform operation l of product g
n:	CF segment index
$C(k)$:	set of indices denoting the linear segments used to approximate the CF for work center k

Decision variables:

Y_{gt}: expected output of product g in period t at the last operation of its routing
I_{gt}: finished goods inventory (FGI) of product g at the end of period t
B_{gt}: backlog of product g at the end of period t
Y_{gtl}: quantity of product g completing its operation l in period t
X_{gtl}: quantity of product g starting operation l in period t
W_{gtl}: WIP of product g at operation l at the end of period t
Z_{gtl}^{k}: fraction of output from work center k allocated to operation l of product g in period t

Parameters:

h_{gt}: unit FGI holding cost for product g in period t
b_{gt}: unit backlog cost for product g in period t
ω_{gt}: unit WIP cost for product g in period t
D_{gt}: demand for product g in period t
α_{gl}: processing time for operation l of product g
β_{k}^{n}: slope of segment n of the CF for work center k
μ_{k}^{n}: intercept of segment n of the CF for work center k.

The ACF model is stated as follows:

$$\min \sum_{g\in G}\sum_{t=1}^{T}\left(\omega_{gt}\sum_{l\in O(g)}W_{gtl}+h_{gt}I_{gt}+b_{gt}B_{gt}\right) \tag{1}$$

subject to

$$W_{g,t-1,l}+X_{gtl}-Y_{gtl}=W_{gtl}, \qquad g\in G, t=1,\dots,T, l\in O(g) \tag{2}$$

$$I_{g,t-1}+Y_{gt}-B_{g,t-1}+B_{gt}-I_{gt}=D_{gt}, \qquad g\in G, t=1,\dots,T \tag{3}$$

$$\alpha_{gl}Y_{gtl}\le \mu_{k}^{n}Z_{gtl}^{k}+\beta_{k}^{n}\alpha_{gl}\left(X_{gtl}+W_{g,t-1,l}\right), \qquad g\in G, t=1,\dots,T, l\in O(g), k\in K(g,l), n\in C(k) \tag{4}$$

$$\sum_{g\in G}\sum_{l\in O(g,k)}Z_{gtl}^{k}=1, \qquad t=1,\dots,T, k\in K \tag{5}$$

$$W_{gtl}, I_{gt}, B_{gt}, X_{gtl}, Y_{gtl}, Z_{gtl}^{k}\ge 0, \qquad g\in G, t=1,\dots,T, l\in O(g), k\in K(g,l). \tag{6}$$

The objective function (1) to be minimized is as the sum of WIP, FGI, and backlog costs over all products, operations, and periods. The WIP balance constraints (2) represent the changes of WIP over time based on the input and output for product g at operation l. The holding of strategic inventory in the wafer fab is not possible. After completion, work is immediately transferred to the next operation in its routing, i.e., we have $X_{gtl}=Y_{g,t,l-1}$. Demand fulfilment and FGI balance are represented by the constraints (3). The CF constraints (4) represent the expected output of each work center as a function of the workload. It consists of the WIP at the beginning and the releases during each period. Output and workload are measured in units of time to allow for varying processing times between operations at the same work center. The Z_{gtl}^{k} decision variables allocate processing capacity to the products and their operations and are limited to a total

of one by constraint set (5). The non-negativity of the decision variables is ensured by constraints (6). We refer to Asmundsson et al. (2009) and Missbauer and Uzsoy (2020) for the details of the ACF formulation.

3.2 VNS Scheme

3.2.1 VNS Principles

VNS is a neighborhood search-based metaheuristic (Mladenovic and Hansen 1997; Hansen and Mladenovic 2001). It is based on the idea to enrich a simple neighborhood search-based method to enable it escaping from local optima. This is carried out by restarting the search for better solution from a randomly chosen neighbour of the incumbent solution. This restarting step, the so-called shaking, is performed using different neighborhood structures of increasing sizes. The basic VNS (BVNS) scheme can be summarized in pseudo code manner as follows:

Initialize: (1) Define k_{max} different neighborhood structures N_k.
(2) Generate an initial solution x.
(3) Initialize $k \leftarrow 1$.
Algorithm: (4) Repeat until stopping criterion is met
(a) Shaking: choose randomly $x' \in N_k$.
(b) Local search: Improve x' by a local search method.
(5) Accept? If x' is better than x, then update $x \leftarrow x'$ and $k \leftarrow 1$, otherwise update the neighborhood structure to be applied by $k \leftarrow (k \bmod k_{max}) + 1$.

Each move in the simulation-based optimization approach for production planning requires at least a single simulation run which tends to be time-consuming. Therefore, we omit the local search Step 4(b) and apply only a reduced VNS (RVNS) scheme.

Next, we have to specify which neighborhood structures are applied in which sequence. This is based on the idea that $\sum_{g\in G}\sum_{t=1}^{T}\left(\omega_{gt}\sum_{l\in O(g)}W_{gtl}+h_{gt}I_{gt}+b_{gt}B_{gt}\right)$ can be decomposed into a timing-related part, i.e. $\sum_{g\in G}\sum_{t=1}^{T}\left(h_{gt}I_{gt}+b_{gt}B_{gt}\right)$, and a part that reflects the congestion behavior of the shop floor, i.e., by $\sum_{g\in G}\sum_{t=1}^{T}\omega_{gt}\sum_{l\in O(g)}W_{gtl}=\sum_{g\in G}\sum_{t=1}^{T}W_{gt}$, where W_{gt} is the total WIP quantity of product g in period t.

3.2.2 Fixed LT-based Neighborhood Structures

We are interested in a neighborhood structure which leads to small changes in the timing-related part of the total cost. We start from the recursive relation for the stock difference:

$$\Delta_{gt} = \Delta_{g,t-1} + Y_{gt} - D_{gt} = I_{gt} - B_{gt}, t = 1, \dots, T \tag{7}$$

for $\Delta_{g0} \coloneqq 0$. Let $X_g \coloneqq \left(X_{g1}, \dots, X_{gt,} \dots, X_{gT}\right)$ be the component for product g of a solution $X \in \mathbb{R}_{+}^{|G|\times T}$ of the production planning problem. We consider an updated solution component $X_g' \coloneqq \left(X_1, \dots, X_{gt} - \Delta_{g,t+L_g}, \dots, X_T\right)$. The output in period $t + L_g$ based on the release in period t is then $Y'_{g,t+L_g} \coloneqq Y_{g,t+L_g} - \Delta_{g,t+L_g}$ if there is enough capacity on the shop floor. Here, we assume an integer-valued fixed LT (FLT) of L_g periods for product g, i.e., we have $Y_{g,t+L_g} = X_{gt}$. Using (7), we obtain $Y_{g,t+L_g} - \Delta_{g,t+L_g} = D_{g,t+L_g} - \Delta_{g,t+L_g-1}$, i.e., we have $\Delta'_{g,t+L_g} \coloneqq \Delta_{g,t+L_g-1} + Y'_{g,t+L_g} - D_{g,t+L_g} = 0$. This means that the resulting backlog and inventory holding costs are zero for period $t + L_g$ if the update quantity is chosen in this way.

Next, we consider the vector of the stock differences $\Delta_g := \left(\Delta_{11}, \ldots, \Delta_{g,t+L_g-1}, \Delta_{g,t+L_g} + \Delta X_{gt} \ldots, \Delta_{gT} + \Delta X_{gt}\right)$ where we must choose $\Delta X_{gt} := -\Delta_{g,t+L_g}$ to obtain $\Delta'_{g,t+L_g} = 0$.

If we want to update the releases of product g for two different periods $t < s$ by means of ΔX_{gt} and ΔX_{gs}, respectively, then by the same argument, the vector of the updated stock differences of product g is

$$\left(\Delta_{g1}, \ldots, \Delta_{g,t+L_g} + \Delta X_{gt}, \ldots, \Delta_{g,s+L_g} + \Delta X_{gt} + \Delta X_{gs}, \ldots, \Delta_{gT} + \Delta X_{gt} + \Delta X_{gs}\right). \quad (8)$$

Hence, to have updated stock differences at periods t and s that are both zero, we must choose $\Delta X_{gt} := -\Delta_{g,t+L_g}$ and $\Delta X_{gs} := -\left(\Delta_{g,s+L_g} + \Delta X_{gt}\right)$. Analogously, considering updating all periods $t = 1, \ldots, T$ and choosing the update quantities as $\Delta X_{gt} := -\left(\Delta_{g,t+L_g} + \sum_{s=1}^{t-1} \Delta X_{gs}\right), t = 1, \ldots, T$, we obtain the updated stock differences as $\left(\Delta_{g1}, \ldots, \Delta_{gL_g}, 0, \ldots, 0\right)$. However, when applying this update procedure to all products and time periods it is likely that the capacity is not enough and hence the FLT assumption is violated. Therefore, we scale the update quantities ΔX_{gt} using parameters $\gamma_{gt} \in [0,1]$ by:

$$\Delta X_{gt} := -\gamma_{gt}\left(\Delta_{g,t+L_g} + \sum\nolimits_{\tau=1}^{t-1} \Delta X_{g\tau}\right). \quad (9)$$

Note that we obtain $\Delta'_{g,t+L_g} = 0$ for $\gamma_{gt} = 1$, and $\gamma_{gt} = 0$ leads to an unchanged release for product g in period t. The new solution has lower timing-related cost under the FLT assumption for the remaining values of the scaling parameters. We define a neighborhood structure N_S^{FLT} for a given set $S \subseteq [0,1]$ by setting

$$N_S^{FLT}(X) := \left\{X' \in \mathbb{R}_+^{|G|\times T} | X'_{gt} := X_{gt} - \gamma_{gt}\left(\Delta_{gt+L_g} + \sum\nolimits_{\tau=1}^{t-1} \Delta X_{g\tau}\right), \gamma_{gt} \in S, g \in G, t = 1, \ldots, T\right\} \quad (10)$$

for a given solution $X \in \mathbb{R}_+^{|G|\times T}$. Note that the elements of $N_S^{FLT}(X)$ can be obtained by sampling scaling parameter matrices $\gamma \in \mathbb{R}_+^{|G|\times T}$ from $S^{|G|\times T}$.

3.2.3 Target Bottleneck Utilization-based Neighborhood Structures

The second class of neighborhood structures deals with reducing WIP cost caused by a solution $X \in \mathbb{R}_+^{|G|\times T}$ in such a way that the desired target bottleneck utilization level obtained from the demand properties is reached. If we execute this solution using a simulation model of the wafer fab, we obtain the realized utilization level of each work center at each period. The utilization level u_t of a work center is obtained as the ratio of the total processing time and the total available time A_t within a period. We obtain $u_t = \sum_{g\in G}\sum_{l\in O(g,k^*)} \alpha_{gl} Y_{gtl}/A_t$ for the utilization of the planned bottleneck work center with $k^* \in K$.

Next, we update the release quantity X_{gt} by ΔX_{gt}. This updates the output quantity approximately to $Y_{gtl} + \Delta X_{gt}$ for all $l \in O(g, k^*)$. The resulting updated bottleneck utilization u'_t can be then approximated by

$$u'_t := u_t + \Delta X_{gt} \sum\nolimits_{l\in O(g,k^*)} \alpha_{gl}/E(A_t), \quad (11)$$

where $E(A_t)$ is the expected available time of operation l of product g in period t. We set $a_{gt} \coloneqq \sum_{l \in O(g,k^*)} \alpha_{gl}/E(A_t)$ for abbreviation. From (11), we directly obtain the update quantity:

$$\Delta X_{gt} = (u'_t - u_t)/a_{gt}\,, \tag{12}$$

where u'_t can be seen as the target bottleneck utilization to be reached after the update. Recall that u_t is the realized utilization level. We define another neighborhood structure N_S^{TU} for a given $S \subseteq [0,1]$ by setting

$$N_S^{TU}(X) \coloneqq \left\{ X' \in \mathbb{R}_+^{|G|\times T} | X'_{gt} \coloneqq X_{gt} + \gamma_{gt}\,(u'_t - u_t)/a_{gt}\,, \gamma_{gt} \in S, g \in G, t = 1, \dots, T \right\} \tag{13}$$

for a given solution $X \in \mathbb{R}_+^{|G|\times T}$. Note that the elements of $N_S^{TU}(X)$ again can be obtained by sampling scaling parameter matrices $\gamma \in \mathbb{R}_+^{|G|\times T}$ from $S^{|G|\times T}$.

3.2.4 Overall Scheme

Initial solutions for the RVNS scheme are computed based on releasing $X_{gt} \coloneqq D_{gt}$. Two RVNS variants are considered. The first one, denoted as RVNS-S, performs only a single simulation run to calculate the total cost value (1), whereas the second variant, abbreviated by RVNS-M, performs multiple independent simulation replications if an improvement of the total cost value is observed for the first replication. The average total cost value is computed based on $m = 5$ independent simulation runs. Appropriate L_g values are determined by simulation experiments (cf. Kacar et al. 2013) assuming a given target bottleneck utilization that is compatible with the given demand.

The entries of the scaling matrices γ are derived as follows. For a given interval S_k, $k = 1, \dots, k_{max}$ we generate $u_{gt} \sim U[S_k]$, where $U[a,b]$ is a continuous uniform distribution over the interval $[a,b]$. Moreover, let b_{gt} be a realization of a Bernoulli distributed random variable with success probability p_k. We then choose $\gamma_{gt} = b_{gt} \cdot u_{gt}$. Note that the success probability of the Bernoulli distributed random variable controls how often $\gamma_{gt} > 0$ is, i.e., an update of a release quantity occurs. A total of 16 pairs (p_k, S_k) is applied, but we show in Table 1 only eight pairs since we have $(p_{2k}, S_{2k}) \coloneqq (p_{2k-1}, S_{2k-1})$ for $k = 1, \dots, 8$.

Table 1. Parameters (p_k, S_k) for neighborhood structures.

k	1	3	5	7	9	11	13	15
p_k	0.1	1.0	0.25	0.25	0.5	0.5	1.0	0.25
S_k	[0.5,1.0]	[0.1,0.3]	[0.0,0.5]	[0.5,0.75]	[0.25,0.5]	[0.5,1.0]	[0.0,0.5]	[−0.25,1.25]

The neighborhood structures are defined for $k = 1, \dots, 16$ as follows:

$$N_k \coloneqq \begin{cases} N_{S_k}^{FLT}, & \text{if } k \equiv 1 \pmod 2 \\ N_{S_k}^{TU}, & \text{if } k \equiv 0 \pmod 2. \end{cases} \tag{14}$$

The $N_{S_k}^{FLT}$ − and $N_{S_k}^{TU}$ −type neighborhood structures are different and one is not a special case of the other due to the different intervals S_k. Therefore, they are not nested. Moreover, the sizes of the neighborhoods are not increasing.

4 IMPLEMENTATION ASPECTS

4.1 Infrastructure for Simulation-based Optimization

The RVNS scheme is coded using the C# programming language. AutoSched AP 11.03 is used as simulation engine. It is a framework based on the C++ programming language. The steering system AutoSched AP calls the executable code of the RVNS scheme. The C# program and AutoSched AP communicate via files. The ACF formulation is coded using again the C++ language, the resulting LPs are solved using the commercial solver ILOG CPLEX 12.7.1. The simulation infrastructure described by Kacar et al. (2013) is reused to execute the production plans from the LP runs. The specified number of lots is released uniformly over the respective periods. All the computational experiments are conducted on a 12th Gen Intel ® Core™ i7-12700 CPU 2.1 GHz PC with 32 GB RAM.

4.2 Simulation Model

The computational experiments are based on the MIMAC I simulation model (Fowler and Robinson 1995) which represents a large-scale wafer fab with more than 200 machines organized in 69 work centers. The steppers of the lithography area serve as a planned bottleneck work center. Batch processing machines and sequence-dependent setup times occur. Exponentially distributed machine breakdowns are the major contributor to variability. In the computational experiments, we use long machine failures as described by Kacar et al. (2013) for the MIMAC I model. Two products are considered in the simulation experiments, each of them requiring over 200 process steps with highly reentrant process flows, i.e., the same work center is visited by a single lot several times. First-In-First-Out (FIFO) dispatching is used. The processing times are deterministic.

5 COMPUTATIONAL EXPERIMENTS

5.1 Design of Experiments

We use a design of experiments similar to the one from Kacar et al. (2013). Time-varying demand is applied. A planning window of $T = 15$ periods is used. The period length is a week. A product mix of 1:1 is considered. Demand D_{gt} that follows a normal distribution $N(\mu_g, \sigma_g^2)$ is used. Here, μ_g is the mean demand of product g that leads to a prescribed bottleneck utilization (BNU) level and $\sigma_g := \mu_g \cdot CV$ is the standard deviation for a given coefficient of variation (CV) value. The μ_g values are chosen for each three-week subinterval in such a way that the given target BNU level is reached. For the scenarios with $BNU = 90\%$, μ_g values leading to BNU=85% or BNU=95% are selected with equal probability. For the scenarios with $BNU = 70\%$, μ_g values resulting in BNU=60% or BNU=80% are selected. Positively correlated demand is considered due to the 1:1 product mix. No demand updates are taken into account. We refer to this demand setting as time-varying load scenarios (cf. Kacar et al. 2013).

Ten different demand realizations are used. Initial WIP values are taken from long simulation runs. For each demand realization, planning is performed with a horizon of 18 periods instead of 15 periods to avoid end of horizon effects. The demand of the additional periods 16-18 is set as the average of the demand of the last three planning periods of the original planning window. The design of experiments is summarized in Table 2.

The CFs from Kacar et al. (2013) are reused. The evaluation of the release schedules obtained by the ACF model is based on 20 independent simulation runs with the horizon of 15 periods (weeks). The final release schedule obtained from simulation-based simulation is assessed in the same way. The unit cost settings $h_{gt} = 15, b_{gt} = 50$, and $\omega_{gt} = 35$ are chosen as in Kacar et al. (2016). We apply a unit revenue value of 180 in the computational experiments. The quantities X_{gt} are released by distributing them evenly within the respective planning period.

Table 2: Design of experiments.

Factor	Level	Count
Planning approaches	ACF, RVNS-S, RVNS-M	3
Demand type	time-varying load	1
Planned bottleneck utilization	70%, 90%	2
CV	0.1, 0.25	2
Machine failure duration	long	1
Demand realizations		10
Simulation replications	1 or 5 per move, 20 for final solution	

We are interested in comparing the performance of the simulation-based optimization approach with the one of the ACF formulation. Moreover, the RVNS-S and RVNS-M variants are compared. The computing time limit of 30 min is used for the RVNS variants, but the obtained total cost and revenue values observed after each five consecutive minutes are also reported. The corresponding time-limited RVNS variants are abbreviated by RVNS-S-c and RVNS-M-c, respectively where c indicates the allowed amount of computing time for the simulation-based optimization.

5.2 Computational Results

The average total cost of the ACF formulation and the RVNS-S and RVNS-M schemes is shown in Figure 1 depending on the allowed computing time limit. The x-axis represents the allowed computing time for the different RVNS variants in minutes and the best solution found within the time limit. The y-axis represents the cost values. The average total cost of 20 independent simulation runs of each final solution is shown. It is worth mentioning that the average total cost values obtained from the ACF formulation are quite similar to results reported by Kacar et al. (2016) for all BNU and CV combinations of time varying (tv) load-type demand. This indicates that the ACF formulation is correctly coded. The results for the low utilization cases found in the upper part of Figure 1 demonstrate that the initial solutions for the RVNS schemes is already fairly competitive to the ones obtained from the ACF formulation. As the computing time limit grows, both the RVNS-S and the RVNS-M outperform the ACF formulation to a large extent with lower deviation in average total cost over different computing time limits.

For both RVNS variants, the largest improvements are obtained within the first five minutes while RVNS-S exhibits a slightly faster convergence. These results can be explained by the observed stability of shop floor under low utilization conditions. There are less CT fluctuations, hence there is capacity required for performing pre- or post-production, and the optimized solutions computed by the ACF formulation yield lower performance. This interpretation supports the superiority of the RVNS-S over the RVNS-M because the limitation of the RVNS-S that it may not be able to escape from local optima with respect to a certain random seed becomes less crucial as there is less randomness caused by the simulation.

At high utilization levels, depicted in the bottom part of Figure 1, the initial solutions obtained for the RVNS are outperformed to a large extent by the ones computed by the ACF formulation. This demonstrates the ability of the ACF formulation to control the congestion at the shop floor and optimize the production quantities over a long horizon under high demand and large shop-floor variability conditions. Yet, the RVNS schemes are able to outperform the ACF formulation as the computing time limit becomes larges. Under the experimental conditions of high utilization and large CV values, however, the RVNS-S requires a longer computing time, i.e. 20 minutes, to outperform the ACF formulation, highlighting the drawback of the RVNS-S scheme to deal with a high uncertainty in an appropriate way. These results lead to the conclusion that the RVNS is able to outperform the ACF formulation as long as the objective function value calculation during planning reflects the real shop-floor randomness. As the randomness at the shop floor increases, the RVNS-S scheme tends to stuck at local optima with respect to a certain random seed, which does not perform well on average over different random seeds. Clearly, one way to overcome this issue is

using the RVNS-M scheme to enhance the accuracy of the objective function value estimation by multiple simulation runs. Thus, there is a trade-off between the allowed computing time and robustness of the simulation-based optimization procedure, increasing the number of independent simulations runs for the objective function calculation increases the former one but also improves the latter one at the same time and vice versa.

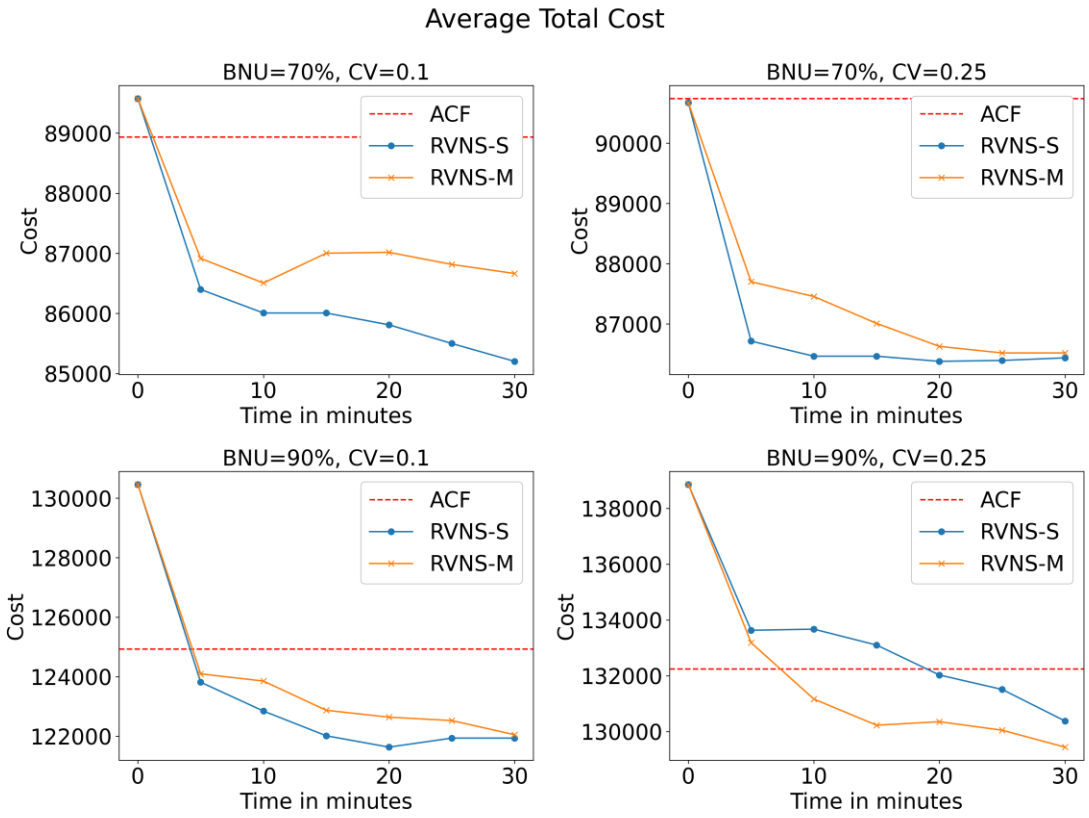

Figure 1: RVNS improvement plots compared to ACF.

The cost breakdowns of the ACF formulation, the RVNS-S, and the RVNS-M, each with a computing time limit of 5 and 30 minutes, respectively, are reported in Table 3. Recall that the concrete value of the symbol c in RVNS-M-c indicates the amount of computing time in minutes. In addition to the total cost, we also report the profit, i.e. the difference of revenue and total cost, in Table 3. The best total cost and profit values for comparable settings, i.e. within a single cell, are always marked bold.

Under low utilization conditions, the RVNS schemes are able to outperform the ACF formulation with respect to total costs and total profit even when a computing time limit of only five minutes is given. Longer allowed computing times enlarge the gap for almost all performance measures. Under high utilization conditions, the better total cost performance of the RVNS schemes is caused by lower WIP cost and FGI holding cost. At high utilization with high CV values, the RVNS scheme with a short computing time limit of five minutes is outperformed by the ACF formulation, where the ACF causes lower backlog cost. It is remarkable that even the RVNS with an allowed computing time of 30 minutes still has worse backlog cost under high bottleneck utilization conditions, and therefore the RVNS schemes, although they have better total cost under all experimental conditions, exhibit lower profit than the ACF scheme. This result states that the cost parameters for simulation-based optimization procedures for production planning should be carefully chosen as one may have better total cost but a slightly smaller profit.

Table 3. Cost breakdowns and profit values.

BNU	CV	Approach	WIP	BO	FGI	Total Cost	Profit
70	10	ACF	79584.6	7105.3	2244.9	88934.7	51455.4
70	10	RVNS-S-5	77911.2	6871.0	1616.8	86399.0	54121.6
70	10	RVNS-M-5	78308.0	6733.0	1874.9	86915.8	53711.0
70	10	RVNS-S-30	77666.1	6134.8	1402.0	**85202.8**	**55334.9**
70	10	RVNS-M-30	79034.9	5461.3	2168.8	86664.9	54040.2
70	25	ACF	79472.6	8687.8	2578.7	90739.0	47538.8
70	25	RVNS-S-5	77239.9	7738.3	1737.2	86715.3	51421.2
70	25	RVNS-M-5	77988.8	7193.0	2518.7	87700.4	50419.9
70	25	RVNS-S-30	77471.1	7226.8	1739.6	**86437.5**	**51629.7**
70	25	RVNS-M-30	77631.9	6723.8	2163.3	86519.0	51538.3
90	10	ACF	109060.5	14372.5	1497.5	124930.6	**50216.7**
90	10	RVNS-S-5	104864.7	17904.3	1046.3	123815.2	47945.3
90	10	RVNS-M-5	100869.7	22042.8	1183.0	124095.4	45497.9
90	10	RVNS-S-30	102461.6	18553.3	920.9	**121935.7**	49229.0
90	10	RVNS-M-30	99662.0	21375.3	1012.8	122050.0	47793.5
90	25	ACF	109792.9	20659.0	1791.6	132243.5	**41794.0**
90	25	RVNS-S-5	106316.5	25910.8	1402.5	133629.8	35981.5
90	25	RVNS-M-5	106319.3	25491.3	1374.5	133185.0	36839.4
90	25	RVNS-S-30	102140.9	27098.0	1139.3	130378.1	37835.5
90	25	RVNS-M-30	100844.1	27373.3	1224.1	**129441.4**	38675.9

6 CONCLUSIONS AND FUTURE WORK

We designed a simulation-based optimization procedure for production planning in wafer fabs. A simulation model of a large-sized wafer fab is applied to assess the proposed method. We observed from the experiments that the simulation-based optimization approach is able to outperform the ACF formulation, a production planning formulation based on nonlinear CFs, under many experimental conditions. However, the simulation-based optimization scheme clearly required more computing time. Overall, it is possible to apply simulation-based optimization for large-scaled wafer fab models.

There are several directions for future research. First of all, the experiments can be repeated in a rolling horizon setting using the martingale model of forecast evolution (MMFE) due to Heath and Jackson (1994) to model demand uncertainty. Another interesting research avenue consists in comparing the gradient-based approach considered by Kacar and Uzsoy (2015) with the simulation-based optimization approach proposed in the present paper. Moreover, we are interested in solving the problem considered by Ziarnetzky and Mönch (2017) with integer-valued decision variables using simulation-based optimization. It is expected that the method proposed in the present paper can also be applied to the strategic network design problem with renewable energy resources studied by Werner et al. (2022). Here, however, one major obstacle is the need to simulate an entire semiconductor supply chain.

ACKNOWLEDGMENTS

This research was supported by the European Commission and the German Authorities through the AIMS5.0 project. The project AIMS5.0 is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Funding Authorities from involved countries under grant agreement no. 101112089. The authors gratefully acknowledge this financial support.

REFERENCES

Asmundsson, J. M., R. L. Rardin, C. H. Turkseven, and R. Uzsoy. 2009. "Production Planning Models with Resources Subject to Congestion". *Naval Research Logistics* 56:142–157.

Fowler, J. W., and J. Robinson. 1995. "Measurement and Improvement of Manufacturing Capacity (MIMAC) Final Report". Technical Report No. Technology Transfer #95062861A, SEMATECH, Austin, Texas.

Gopalswamy, K., and R. Uzsoy. 2019. "A Data-driven Iterative Refinement Approach for Estimating Clearing Functions from Simulation Models of Production Systems". *International. Journal of Production Research* 57(19): 6013–6030.

Gopalswamy, K., and R. Uzsoy. 2020. "Conic Programming Reformulations of Production Planning Problems". *European Journal of Operational Research* 292(3):953–966.

Hansen, P., and N. Mladenovic. 2001. "Variable Neighborhood Search: Principles and Applications". *European Journal of Operational Research* 130:449–467.

Heath, D., and P. Jackson. 1994. "Modeling the Evolution of Demand Forecasts with Application to Safety Stock Analysis in Production/Distribution Systems". *IIE Transactions* 26:17–30.

Kacar, N. B., L. Mönch, and R. Uzsoy. 2013. "Planning Wafer Starts using Nonlinear Clearing Functions: a Large-Scale Experiment". *IEEE Transactions on Semiconductor Manufacturing* 26(4):602–612.

Kacar, N. B., L. Mönch, and R. Uzsoy. 2016. "Modeling Cycle Times in Production Planning Models for Wafer Fabrication". *IEEE Transactions on Semiconductor Manufacturing* 29(2):153–167.

Kacar, N. B., and R. Uzsoy. 2015. "Estimating Clearing Functions for Production Resources Using Simulation Optimization". *IEEE Transactions on Automation Science and Engineering* 12(2):539–552.

Leachman, R. 2001. "Semiconductor Production Planning". In: *Handbook of Applied Optimization*, edited by P. Pardalos and M. Resende, 746–762. New York: Oxford University Press.

Liu, J., C. Li, F. Yang, H. Wan, and R. Uzsoy. 2011. "Production Planning for Semiconductor Manufacturing via Simulation Optimization". In *2011 Winter Simulation Conference (WSC),* 3617–3627 https://doi.org/10.1109/WSC.2011.6148055.

Missbauer, H., and R. Uzsoy. 2020. *Production Planning with Capacitated Resources and Congestion*. New York: Springer.

Missbauer, H. 2020. "Order Release Planning by Iterative Simulation and Linear Programming: Theoretical Foundation and Analysis of its Shortcomings". *European Journal of Operational Research* 280(2):495–507.

Mladenovic, N. and P. Hansen, P. 1997. "Variable Neighborhood Search". *Computers & Operations Research* 24:1097–1100.

Mönch, L., J. W. Fowler, and S. J. Mason. 2013. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*. New York: Springer.

Mönch, L., R. Uzsoy, and J. W. Fowler. 2018. "A Survey of Semiconductor Supply Chain Models Part III: Master Planning, Production Planning, and Demand Fulfillment". *International Journal of Production Research* 56(13):4524–4545.

Völker, T., and L. Mönch. 2023. "Data-driven Production Planning Models for Wafer Fabs: An Exploratory Study". *IEEE Transactions on Semiconductor Manufacturing* 36(3):445–457.

Werner, M., L. Mönch, and J.-Z. Wu. 2022. "A Planning Model for Incorporating Renewable Energy Sources into Semiconductor Supply Chains". In *2022 Winter Simulation Conference (WSC)*, 3465-3476 https://doi.org/10.1109/WSC57314.-2022.10015505.

Zhang, Z., Y. Gong, and Z. Guan. 2022. "Clearing Function-based Simulation Optimization for Release Planning Under Digital Twin Wafer Fab". *IFAC PapersOnLine* 55-10: 2539–2544.

Ziarnetzky, T., and L. Mönch. 2016. "Simulation-based Optimization for Integrated Production Planning and Capacity Expansion Decisions". In *2016 Winter Simulation Conference (WSC)*, 2992–3003 https://doi.org/10.1109/WSC.2016.7822333.

Ziarnetzky, T., L. Mönch, T. Kannaian, and J. Jimenez. 2017. "Incorporating Elements of a Sustainable and Distributed Generation System into a Production Planning Model for a Wafer Fab". In *2017 Winter Simulation Conference (WSC)*, 3519–3530 https://doi.org/10.1109/WSC.2017.8248066.

AUTHOR BIOGRAPHIES

HAJIME SEKIYA is currently a PhD student in information systems at the University of Hagen. He received a master degree in applied mathematics from the Technical University of Munich. His research interests are production planning for semiconductor manufacturing, order release, and discrete-event simulation. His e-mail address is hajime.sekiya@fernuni-hagen.de.

LARS MÖNCH is full professor of Computer Science at the Department of Mathematics and Computer Science, University of Hagen where he heads the Chair of Enterprise-wide Software Systems. He holds M.S. and Ph.D. degrees in Mathematics from the University of Göttingen, Germany. After his Ph.D., he obtained a habilitation degree in Information Systems from Technical University of Ilmenau, Germany. His research and teaching interests are in information systems for production and logistics, simulation, scheduling, and production planning. His email address is Lars.Moench@fernuni-hagen.de.