Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

SUPPLY CHAIN DIGITAL TWIN FRAMEWORK FOR HYBRID MANUFACTURING STRATEGY SELECTION: A CASE STUDY FROM THE SEMICONDUCTOR INDUSTRY

Amir Ghasemi[1], Sanja Lazarova-Molnar [1,2], and Cathal Heavey[3]

[1]Institute AIFB, Karlsruhe Institute of Technology, Karlsruhe, GERMANY
[2]Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, DENMARK
[3] CONFIRM Research Centre, School of Engineering, University of Limerick, Limerick, V94 T9PX, IRELAND

ABSTRACT

Within the Manufacturing Supply Chain planning domain, the integration of Digital Twins as a decision-making tool presents a promising development path. This research introduces a novel Supply Chain Digital Twin (SCDT) framework for manufacturing supply chains, specifically tailored to address the manufacturing strategy selection problem at the strategic product level. To demonstrate our proposed SCDT framework, we employ the Business Process Model and Notation (BPMN) as a model-based systems engineering tool. The primary aim of this study is to detail the design and integration of SCDTs within manufacturing supply chain networks, facilitating decisions on manufacturing strategy selection. The practical applicability of our proposed SCDT framework is further demonstrated through a case study in the semiconductor industry, highlighting its utility and potential benefits.

1 INTRODUCTION

Digital Twins (DTs) stand as one of the main outcomes of Industry 4.0 technologies, particularly within the domain of manufacturing systems digitization (Ghasemi et al. 2024). A DT is a virtual replica of a physical entity or system that uses real-time data and advanced simulation to optimize performance, predict outcomes, and enhance decision-making throughout the asset's life-cycle (San 2021). SCDTs are a class of DTs that mirror physical SC processes, allowing companies to simulate, analyze, and optimize their operations in nearly real-time for improved decision-making and resilience.

The concept of SCDT, for the first time, was introduced in the conceptual model proposed by Ivanov and Dolgui (2021). The authors refer to SCDT as a digital model that reflects the states of a SC network at any specific instant in real time. In another study, Park et al. (2021) proposed a distributed DT for dynamic SC control in Make-To-Order (MTO) manufacturing environments, providing detailed architectural frameworks and operational procedures for resilience. However, their research lacks detailed process planning for personalized production and functionalities for demand forecasting and inventory management. In another research, Maheshwari et al. (2023) proposed a DT for enhancing food SC resilience, detailing technical functionalities and applications for integrated Procurement, Production, and Distribution (PPD) strategies. However, the authors primarily outlined conceptual frameworks, indicating a need for further empirical research on dynamic procurement, delivery strategies, and the integration of complex data sets for real-world implementation. Review articles on SCDT highlight the limited number of research and the need for architecture definition within this domain (Lugaresi et al. 2024). Accordingly, despite the growing interest in leveraging SCDTs for improved planning within Manufacturing Supply Chain (MSC) networks, the practical and empirical research in this domain remains limited. Moreover, there is a significant need for standardization to guide future research efforts in this rapidly evolving domain.

A DT serves multiple purposes in MSC planning, offering extensive capabilities to enhance operational efficiency and decision-making. In this paper, we concentrate on one specific application aiming to demonstrate how a SCDT can be effectively developed and used. Specifically, we focus on its application in dynamically generating hybrid Manufacturing Strategy Selection (MSS) for MSCs.

From the MSC planning perspective, hybrid MSS emerges as a crucial facilitator of resilience and agility (Beemsterboer et al. 2016). Hybrid MSS merges the predictive production capabilities of Make-To-Stock (MTS) with the responsive flexibility of Make-To-Order (MTO) strategies. MTS strategy involves producing products in anticipation of future demand and storing them in inventory, while MTO entails manufacturing products only after receiving specific customer orders. This synthesis in hybrid MSS enables the system to efficiently balance inventory levels with customer demand variability, enhancing their ability to swiftly adapt to market changes while optimizing operational performance indicators. Despite its strategic importance, the domain of hybrid MSS is marked by a significant research gap, particularly in its practical deployment and the exploration of its full potential within diverse manufacturing environments and in leveraging advanced Industry 4.0 decision-making tools, such as DT, for the automated generation of MSS decisions (Peeters and Van Ooijen 2020).

The semiconductor SCs are fundamental to the global economy, driving innovation and growth across industries by enabling advanced technologies that are critical for competitiveness and economic development in the digital age. Semiconductors are an integral part of modern life, with their use ranging from everyday consumer electronics to critical infrastructure systems, showcasing their essential role in the digital world around us (Mönch et al. 2017; Ghasemi et al. 2018; Ghasemi et al. 2018). Thus, enhancing the planning efficiency in semiconductor SCs is crucial for ensuring the stability and continuity of global digital-dependent world. There are studies, including those by Forstner and Monch (2013), Arima et al. (2016), that have explored the implementation of hybrid MSS in semiconductor SCs. However, due to the complexity of SC planning in semiconductor industry, there remains a gap in applying advanced tools such as SCDTs to address such decision-making problems within the sector (Wagner et al. 2019).

In response to the identified gaps, this paper presents a preliminary SCDT framework tailored for hybrid MSS in global, multi-plant MSC environments. To illustrate the potential applications of this framework, we demonstrate its functionality through a simplified case study in the Semiconductor SC.

The remainder of this paper is organized as follows: Section 2 details the proposed SCDT. Section 3 presents the experiment design including a case study of a semiconductor SC scenario to illustrate the application and effectiveness of the proposed framework. Section 4 discusses the results and implications of the study, highlighting key findings and their practical relevance. Finally, Section 5 concludes the paper with a summary of the contributions, limitations of the current study, and suggestions for future research directions.

2 PROPOSED SUPPLY CHAIN DIGITAL TWIN FRAMEWORK

The BPMN diagram shown in Figure 1 is created to clearly show the architecture of SCDT framework proposed by this research. BPMN is a graphical representation that outlines business processes in a workflow, enabling a clear visualization of step-by-step operations and interactions within and between organizations. BPMN's notation is designed to be easily understood by all business stakeholders, including non-technical users (Chinosi and Trombetta 2012). BPMN's potential for describing how a SCDT is positioned within a MSC planning framework lies in its ability to map the complex relationships and data flows between the SCDT and various SC Information Technology (IT) components. Note that our main objective in this paper is to highlight the design and embedding structure of SCDTs within MSC networks to generate MSS decisions. Accordingly, in this section, we elaborate on the supply chain planning modules to a degree that allows us to clearly outline the architecture of the proposed SCDT framework. In other words, SC planning processes are usually very complex even in the case of small SC networks, thus showing all details in one graph is impossible. This section presents the proposed SCDT from a high-level perspective using the BPMN diagram in Figure 1 and Algorithm 1. In the following sections we will focus on SCDT (bottom box of Figure reffig:Designed SCDT framework in BPMN language.), placing emphasis on how an SCDT could be used, by describing, at a high level: (i) Data Infrastructure, (ii) Online Simulation Core and (iii)

Decision Making Core. The proposed SCDT lies in the category of autonomous twin since it combines Simulation, Optimization, AI, and automation to learn from data dynamically and generate optimal MSS decisions.

2.1 Data Infrastructure

This component of SCDT is responsible for managing data transactions from and/or to SC physical and IT systems. This includes Manufacturing Execution, Warehouse Management, MRP (Material Requirements Planning), SC financial, and demand planning systems. Both Manufacturing Execution System (MES) and Warehouse Management System (WMS), installed on the physical side of a MSC network, oversee the management of daily transactional and operational data activities. Moreover, they act as a bridge between physical side of an SC (e.g., manufacturing) and SC IT systems (i.e., DT, ERP, and APS systems). As shown in Figure 1, in the proposed SCDT framework, MES and WMS produce two sets of transactional data to be consumed by the SCDT: Work In Progress (WIP) and Inventory Data, respectively. To generate these transactional data, both systems take scheduled snapshots from the current status of physical manufacturing and logistics resources (e.g., machines and warehouse shelves, respectively). Therefore, a shorter interval between two snapshots enables the transactional data to describe the physical system with greater real-time accuracy. As shown in Figure 1, these transactional data are captured by the SCDT and stored in a Transactional Database (Transactional DB). There are several technologies available for deploying Transactional DBs such as SQL, NoSQL, In-Memory, and Blockchain-based DBs. Note that describing the technologies for designing DBs are out of the scope of this research (for further details on transactional databases, we direct readers to Kumar and Singh (2023)). As illustrated in Figure 1, the proposed SCDT leverages Master Data sourced from the DT Master Data DB. This Master Data is aggregated through pipelines from existing ERP and APS systems. The ERP system usually hosts data pertinent to finances, resources, demands, and Material Requirements Planning (MRP), while APS systems enhance supply chain planning by providing sophisticated decision-making capabilities. Specifically, in this research, APS contributes demand forecast data and delivery plans. These datasets are then aggregated via data pipelines from both systems and stored in the Master Data Database. It is important to note that, unlike transactional data, this Master Data is collected less frequently, as it does not undergo rapid changes over short intervals such as daily. Note that both Update DT Master Data DB and Update Transactional Data DB modules are responsible for ensuring data quality by employing required Data Cleaning and Data Validation procedures.

2.2 Online Simulation Core

In the proposed SCDT, the Online Simulation Core (see bottom diagram of Figure 1) is designed to extract a simulation model that reflects the current status of the MSC network based on the data provided by the Data Infrastructure. Moreover, the Online Simulation Core checks for updates in the Data Infrastructure and, if necessary, updates the core simulation model.

The pseudo code in Algorithm 1 provides a high-level outline of the structure and flow of the Core Simulation Model. The Core Simulation Model processes various types of input data to simulate the current status of an MSC. The model takes MRP, Demand, Financial, Online WIP, and Online Inventory Status Data provided by the Data Infrastructure as inputs. Through a procedure named *GenerateSimulationModel*, the Core Simulation Model initializes a simulation model and then sequentially processes each type of input data. Finally, the Simulation Snapshot DB is updated by storing the historical Core Simulation Data. Functions such as *ProcessMRPData*, *ProcessDemandForecast*, *ProcessFinancialData*, *ProcessOnlineWIPData*, and *ProcessOnlineInventoryStatus* are dedicated to analyzing their respective data types. They assess MRP, demands, costs and revenue implications, current production status, and current stock levels to update the simulation model accordingly. Following the data processing, the *IntegrationOfData* function combines all analyzed information into integrated data. Standardization is crucial for accurate data integration. Specifically, the structure of both the inputs and outputs of *IntegrationOfData* should be standardized

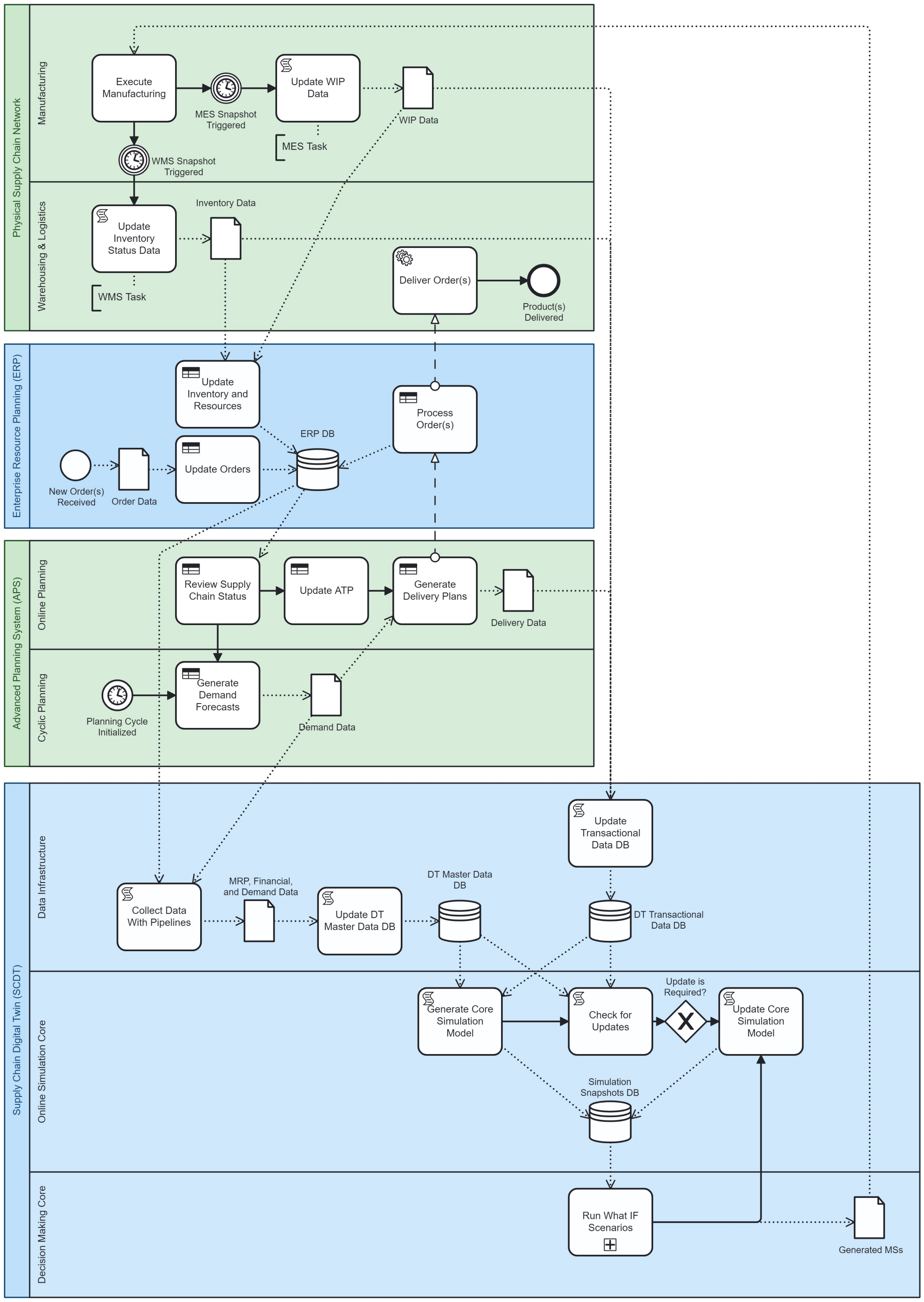


Figure 1: Designed SCDT framework in BPMN language.

Algorithm 1 Online Simulation Core

```
1: Inputs: MRP Data, Demand Data, Financial Data, Online WIP Data, Online Inventory Data
2: Output: Simulation Model (of the Current Supply Chain Network Status)
3: procedure GenerateSimulationModel(MRP Data, Demand Data, Financial Data, Online WIP Data,
   Online Inventory Data)
4:     Initialize Simulation Model
5:     ProcessMRPData(MRP Data)
6:     ProcessDemandData(Demand Data)
7:     ProcessFinancialData(Financial Data)
8:     ProcessOnlineWIPData(Online WIP Data)
9:     ProcessOnlineInventoryStatus(Online Inventory StatusData)
10:    IntegrationOfData() # Generates the Integrated Data.
11:    UpdateSimulationSnapshotDB(Integrated Data)
12:    SimulationModel $\leftarrow$ CreateSimulation(Integrated Data)
13:    return Simulation Model
14: end procedure
```

using consistent data semantics. Next, the *UpdateSimulationSnapshotDB* function pushes the integrated data into the Simulation Snapshot DB. Finally, the integrated data is utilized by the *CreateSimulation* function to simulate the current supply chain network status. This simulation model aims to provide an up-to-date snapshot of the supply chain's current state, incorporating financial, demand, production, and inventory considerations. Depending on the use case, the *CreateSimulation* function is capable of producing highly visual dashboards utilizing tools such as Microsoft Power BI (Becker and Gould 2019), or alternatively, it can generate just a spreadsheet hosted on the cloud for representation.

2.3 Decision Making Core

In SCDT, the Decision Making Core (see bottom diagram of Figure 1) is responsible to provide decisions. This is usually referred to as running "what if" scenarios to obtain best decisions according to predefined performance measures. A variety of techniques are available for running "what if" scenarios in SCDTs, including Simulation Optimization (SO), AI-driven methods, and traditional techniques such as linear programming, each chosen based on the specifics of the application domain. Among them, SO has proven to be particularly effective for decision-making in complex environments (Ghasemi et al. 2021). Thus, here, we leverage SO techniques to execute "what if" scenarios for generating hybrid MSS decisions dynamically. Implemented SO technique utilizes historical data drawn from previous instances of the Core Simulation, which are saved in the Simulation Snapshots DB. Moreover, it incorporates the current status of the MSC network using the latest snapshot in the same DB. In our proposed SCDT framework, optimal MSS decisions are calculated dynamically. As shown in Figure 1, Run What IF Scenarios is an aggregated process in the proposed SCDT. In this research, we use an SO algorithm to run What IF Scenarios aiming to generate optimal optimal MSS decisions. In the following, we detail the architecture of developed SO.

2.3.1 Simulation Optimization to Generate Optimal Manufacturing Strategy Selection

Our SO algorithm integrates a Genetic Algorithm (GA) with a simulation model to optimize MSS decisions for each product in a MSC framework. As outlined in Algorithm 2, SO processes through two input categories. The first category involves historical MSC data sourced from the Simulation Snapshots DB, which encapsulates demands, plant delivery Lead Times, plant Capacities in each period, and the aggregated number of products. The second category comprises SO settings, adjusted by the user, which include variables such as the Number of Past Periods to take into account, the range of possible MS Options per Product, and GA parameters, specifically, the Number of Generations, Population Size, and Mutation Rate.

Product	**1**	**2**	**3**	**4**	**5**
MS	1	2	4	3	2

Figure 2: An example of a chromosome in GA of SO1.

In the SO that we implemented, the procedural execution begins with the *InitializeGA* function, which seeds the initial population for the GA based on the Demands, Plant Capacities, Lead Times, and user-defined SO Settings. Figure 2 illustrates the design of a chromosome (MSS solution) within the proposed GA. Accordingly, the length of a chromosome equals to the total number of products, where each gene refers to the selected MS for the corresponding product. For instance, in Figure 2, Product three follows the fourth MS (MS = 4). In this research, we consider five MS options for each product including one MTO and four MTS strategies. Finally, the selected MS for each product defines the amount of manufacturing orders in each planning period. Given the following notations:

- $D_h(p,t)$: Historical demand (Actual Demand) for product p in period t.
- $D_e(p,t)$: Estimated demand for product p in period t, provided by the planning department.
- CAP: Capacity of the selected manufacturing plant.
- τ: Lead times for each node.
- α: Smoothing constant used in exponential smoothing forecasting.
- $Q(p,t)$: Manufacturing order quantity for product p in period t.

The manufacturing order quantity for each product, based on its MS, is calculated as follows:

MTO Product

$$Q(p,t) = \min\left\{\frac{D_h(p,t-1)+D_e(p,t-1)}{2}, CAP\right\} \tag{1}$$

MTS1 Product

MTS with averaging demands:

$$Q(p,t) = \min\left\{\frac{1}{t}\sum_{i=1}^{t}\frac{D_h(p,i)+D_e(p,i)}{2}, CAP\right\} \tag{2}$$

MTS2 Product

MTS with linear regression:

$$Q(p,t) = \min\left\{\max\left(\hat{D}_{\text{LR}}\left(\frac{D_h(p,i)+D_e(p,i)}{2}, t\right), 0\right), CAP\right\} \tag{3}$$

where $\hat{D}_{\text{LR}}$ represents the forecasted demand using linear regression.

MTS3 Product

MTS incorporating safety stock:

$$Q(p,t) = \min\left\{\max\left(\overline{D}\left(\frac{D_h(p,i)+D_e(p,i)}{2}, t\right) + z\cdot\sigma\cdot\sqrt{\tau}, 0\right), CAP\right\} \tag{4}$$

where $\overline{D}$ and σ represent the average and standard deviation of the averaged demands, respectively.

MTS4 Product

MTS with exponential smoothing:
for $t > 1$:

$$Q(p,t) = \min\left\{\max\left(\alpha \cdot \frac{D_h(p,t-2)+D_e(p,t-2)}{2} + (1-\alpha)\cdot\frac{D_h(p,t-1)+D_e(p,t-1)}{2}, 0\right), CAP\right\} \tag{5}$$

For $t = 1$, the demand is taken as the average of historical and estimated demands directly.

Algorithm 2 SO

```
1: Inputs:
2:    Simulation Snapshots DB Data: Demands, Capacities, Lead Times
3:    SO Settings modified by the User: Number of Past Periods, MSS Options per Product, GA Parameters
   (Number of Generations, Population Size, Mutation Rate)
4: Outputs: Optimal MSS for each Product, Best Fitness Score
5: procedure SO(Demands, Plant Capacities, Lead Times, SO Settings)
6:     Population $\leftarrow$ InitializeGA(Demands, Plant Capacities, Lead Times, SO Settings)
7:     for each generation in Number of Generations do
8:         FitnessScores $\leftarrow$ EvaluateFitness(Population)
9:         Population $\leftarrow$ EvolvePopulation(Population, FitnessScores)
10:    end for
11:    Optimal MSS for each Product, Best Fitness Score $\leftarrow$ FindBestSolution(Population)
12:    return Optimal MSS for each Product, Best Fitness Score
13: end procedure
```

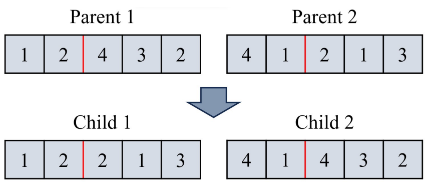
After generating the initial population of solutions (chromosomes), the GA algorithm advances through successive generations, as prescribed by the user's settings, where the *EvaluateFitness* function (see Algorithm 2) assesses each candidate MSS decision's fitness using the simulation model. Considering the Number of Past Periods defined by the user, the simulation model calculates the net profit corresponding to each chromosome using the following formula adapted from Forstner and Monch (2013). The net profit for a product p in period t is calculated as follows:

$$\text{Net Profit}_{p,t} = \underbrace{\text{rev}_{p,t} \times \text{U}_{p,t}}_{\text{Revenue}} - \underbrace{\left((\text{b}_{p,t} \times \text{S}^B_{p,t}) + (\text{h}_{p,t} \times \text{I}_{p,t}) + (\text{c}_{p,t} \times \text{M}_{p,t}) + (\tilde{\text{c}}_{p,t} \times \text{U}_{p,t})\right)}_{\text{Costs}} \tag{6}$$

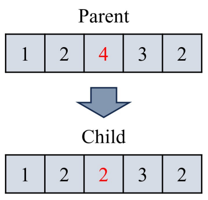
where:

- $\text{rev}_{p,t}$ is the expected revenue per product p in period t.
- $\text{U}_{p,t}$ is the number of sold items from product p in period t.
- $\text{b}_{p,t}$ is the cost due to unmet demand of one unit of product p postponed from period t to $t+1$.
- $\text{S}^B_{p,t}$ is the backlog of demand for product p at the end of period t.
- $\text{h}_{p,t}$ is the inventory costs for holding one item of product p within period t.
- $\text{I}_{p,t}$ is the inventory level of product p at the end of period ts.
- $\text{c}_{p,t}$ is the manufacturing cost per unit of product p in period t.
- $\text{M}_{p,t}$ is the number of items of product p that are in WIP in period t.
- $\tilde{\text{c}}_{p,t}$ is the overall processing costs that are charged if one item of product p is delivered in period t.

Based on the number of considered past periods, the simulation model calculates the total accumulated revenue using Equation (6) for each chromosome. Algorithm 3 outlines the simulation model designed to calculate cumulative revenue based on a set of inputs, including a Chromosome, the Number of Past Periods to be considered, Number of Products, Lead Times, Number of Manufacturing Nodes, and Demands. It starts by initializing the central inventory, WIP for each manufacturing node, and backlogs. The core of the simulation, the *SimulatePeriod* procedure, calculates the WIP and total revenue for each period. For every manufacturing node, *SimulatePeriod* processes orders in the WIP, updating the central inventory and recording deliveries for orders due in the current period. It then adjusts inventory and backlogs based on the deliveries and demands for each product. The *UpdateWIP* function then updates WIP based on the given Chromosome, placing orders to manufacturing nodes using Equations (1) to (5). The simulation runs through all specified past periods up to the current period, continually updating the cumulative revenue based on the revenue generated in each period, ultimately returning the total accumulated revenue for a chromosome (solution). After generating the initial population and calculating the fitness value for each solution, the *EvolvePopulation* (see Algorithm 2) function dynamically refines the population of solutions through selection, crossover, and mutation strategies, steering the population towards optimal solutions. The selection process employs a roulette wheel approach, where the probability of a chromosome being selected as a parent (for crossover) is proportional to its fitness relative to the total fitness of the population. This method ensures that chromosomes with higher fitness have a greater chance of contributing their genes to the next generation. The crossover procedure, specifically a single-point crossover, combines genes from two parent chromosomes at a random point, producing offspring that inherit a mix of parental traits (see the illustrative example in Figure 3a). This mechanism introduces new genetic combinations into the population, facilitating the exploration of the solution space. Mutation adds further diversity by randomly altering genes in a chromosome according to a predefined mutation rate (see the illustrative example in Figure 3b). This step prevents the algorithm from converging prematurely on local optima by ensuring a continuous introduction of random genes. After mutation, the offspring's fitness is calculated using the simulation model (similar to the fitness calculation procedure discussed above). The generation's evolution concludes by merging the new chromosomes with the existing ones and selecting the fittest individuals to form the next generation. After finding the best MS for each product (Optimal MSS in Algorithm 2), they are communicated to both Online Simulation Core of SCDT (to update the core simulation model with the new MSS decisions) and the physical side of MSC (to produce products with updated quantities based on optimal MSs).



(a) An example of used Crossover.



(b) An example of used mutation.

Figure 3: GA operators in developed SO.

3 EXPERIMENT DESIGN

The primary objective of the experiments in this research is to showcase a specific use of SCDT for MSCs, specifically focused on identifying the optimal MSS for each product. In this section, we generate a semiconductor SC test environment. We use a simulation model similar to the one describe above (see Algorithm 3) as a Virtual Semiconductor SC (VSSC) to generate required data for the SCDT (similar to the SCDT architecture provided in Figure 1). To generate VSSC parameters, we partially use the semiconductor SC experiment design proposed by Forstner and Monch (2013). In the following, the main assumptions within the designed VSSC are detailed.

Algorithm 3 Simulation Model

```
1: Inputs: Chromosome, Number of Past Periods, Number of Products, Lead Times, Number of Manufacturing Nodes, Demands
2: Output: Cumulative Revenue
3: Initialize central inventory, WIP for each Manufacturing Node, backlogs, cumulative revenue, inventory and backlog costs.
4: procedure SimulatePeriod(period, WIP, Chromosome, Number of Products, Number of Manufacturing Nodes, Demands)
5:     Reset deliveries for each product to zero.
6:     for each node in Number of Manufacturing Nodes iterating through WIP do
7:         for each order in node's WIP do
8:             if order's due period is the current period then
9:                 Update central inventory and record delivery.
10:                Remove delivered order from WIP.
11:            end if
12:        end for
13:    end for
14:    for each product in Number of Products do
15:        Update inventory, backlogs, and revenue according to deliveries and Demands.
16:    end for
17:    UpdateWIP(WIP, Chromosome, Number of Products, Number of Manufacturing Nodes, Demands)
18:    return period's revenue and WIP.
19: end procedure
20: for period in Number of Past Periods do
21:     Run SimulatePeriod and update the accumulative revenue.
22: end for
23: return Total accumulated revenue.
```

In VSSC, we consider the frontend side of a semiconductor SC with three frontend fabs (manufacturing nodes). Each frontend fab has specific location, capacity, and lead time (cycle time) in delivering products. The product here refers to finished dies that are created from raw wafers through multiple production steps in each frontend fab. We consider 8 different final products with their specific features described in Table 1. VSSC generates demands with four different patterns: Constant, Trend, Cyclical, and Random. Algorithm 4 details the calculations to generate each demand pattern. Moreover, $\mathrm{h}_{p,t}$, $\mathrm{b}_{p,t}$, $\mathrm{c}_{p,t}$, and $\tilde{\mathrm{c}}_{p,t}$ values are equal to $0.2/52 \times \mathrm{rev}_{p,t}$, $5 \times \mathrm{h}_{p,t}$, $0.1/52 \times \mathrm{rev}_{p,t}$, $0.8 \times \mathrm{rev}_{p,t}$, respectively. We consider a total of 36 periods, with the current period set at 12. During the initial 12 periods, the VSSC assigns random MS values for each product based on the generated demand values and specified MSC settings, updating WIP, costs, and revenues accordingly. From period 13 onwards, VSSC transmits all accumulated data to the SCDT for the generation of optimal MSS decisions. This setup enables the SCDT to utilize data from the first 12 periods to inform and refine MSS decisions dynamically as new data becomes available from period 13 onwards. For frontend fabs one, two, and three, the production capacities are 8,000, 8,000, and 10,000 wafers per period, respectively. Additionally, the delivery lead times for orders at frontend fabs one, two, and three are two, one, and two periods post-order placement, respectively. Regarding SO parameters, we utilize 50 generations for the GA, a population size of 40, and a mutation rate of 0.2. Finally, to run experiments we used Google Colab, which provided us with 12.7 GB of RAM and 107.7 GB of disk capacity.

Table 1: Product features.

Product	P1	P2	P3	P4	P5	P6	P7	P8
$rev_{p,t}$	10.4	11.3	8.7	17.4	19	16.2	19.2	18.8
Demand Pattern	1	1	2	2	3	3	4	4

Algorithm 4 Demand Generation in VSSC

```
1: Inputs: Demand Pattern, current_month, base_demand, Number of Periods
2: Output: Demand
3: procedure GenerateDemand(Demand Pattern, current_month, base_demand, Number of Periods)
4:     Initialize demand with base_demand
5:     if Demand Pattern == “constant" then
6:         demand ← demand + 0
7:     else if Demand Pattern == “trend" then
8:         demand ← demand + current_month × 100
9:     else if Demand Pattern == “cyclical" then
10:        demand ← demand + base_demand × 0.3 × cos(current_month / Number of Periods × 2 × pi) + 100
11:    else if Demand Pattern == “random" then
12:        demand ← demand + random_int(-base_demand × 0.2, base_demand × 0.2)
13:    end if
14:    demand ← max(0, demand) (Ensure non-negative demand)
15:    return round(demand) (Ensure integer demand)
16: end procedure
```

4 EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we present the outcomes of our experiments and analyze the implications of the findings. As discussed above, following the operation of the VSSC for the initial 12 periods, we employ the SCDT from period 13 through 36 to generate optimal MSs and subsequently feed these back into the VSSC. Moreover, we replicate the VSSC and SCDT integration 10 times. This leads to $(36-13)\times 10 = 240$ SCDT executions. Table 2 shows the occurrence percentage of MS in the optimal decisions provided by the SCDT to the VSSC across all 240 transactions for all products.

Table 2: Occurrence percentage of MS in SCDT feedbacks for each product.

P1	P2	P3	P4	P5	P6	P7	P8
MTS4 (96%) MTS1 (4%)	MTS4 (95%) MTS1 (5%)	MTS3 (98%) MTS1 (2%)	MTS3 (100%)	MTS3 (100%)	MTS3 (100%)	MTO (100%)	MTO (98%) MTS3 (2%)

Based on the obtained results, the optimal MS used by the SCDT for the products P1 and P2 with their constant demand patterns is mainly MTS4, which is a MTS with exponential smoothing. Products P3 to P6, ranging between trend and cyclical demand patterns, are mainly assigned to MTS3 MS, which is a MTS that incorporates safety stock. In contrast, products with random demands (P7 and P8) show a higher inclination towards MTO MS. The conducted experiments offer two primary insights. First, they demonstrate the capability of SCDT in making optimal MSS decisions dynamically and across various planning periods. SCDT effectively responds to fluctuating demand patterns, which is especially critical in the context of supply chain resiliency. This dynamic adaptability ensures that decision-making processes can adjust to changes in market behaviors in real-time. Second, the experiments highlight the relationship between demand patterns and the optimal MSS decision. Specifically, the findings indicate that for random demand patterns, the MTO MSS approach proves to be superior. Conversely, for other demand patterns, different MTS MSS approaches are selected, emphasizing the tailored adaptability of the SCDT system to various market conditions.

5 CONCLUSIONS AND FUTURE WORK

In this study, we introduce a Supply Chain Digital Twin (SCDT) framework aimed at facilitating Manufacturing Strategy Selection decisions within Manufacturing Supply Chains. We used Business Process Model and Notation (BPMN) language to present and standardize the SCDT framework (see Figure 1). We demonstrated SCDT's application through a semiconductor industry case study. The results from this case study highlight the SCDT's capability to dynamically generate MSS decisions across different planning periods.

Although this paper details various components of the proposed SCDT, it also opens up numerous avenues for further research and exploration. Currently, our research focuses on optimizing MSS decisions for MSCs, leaving other critical Supply Chain-level decisions, such as plant order allocation and master scheduling, as potential areas for future application of SCDTs. Moreover, in this research, we focused on demand fluctuations as a primary source of variability. However, in practice, there are several network-level uncertainties within MSCs that need further exploration. Additionally, the current use of a simplistic Single-Objective Simulation Optimization algorithm in our SCDT could be substantially improved by integrating more sophisticated, AI-driven techniques, including Machine Learning models. Such advancements could enhance the predictive capabilities of corresponding SCDTs, thereby enriching their decision-making effectiveness. Another promising research direction involves development and refinement of validation procedures for SCDTs. Specifically, validation processes for each component of the SCDT framework (including data and simulation model validations) should be thoroughly examined and enhanced.

REFERENCES

Arima, S., H. Bu, Y. Ye, K. Kitamoto, and K. Shimada. 2016. "Optimal Production and Capacity Planning for Make-to-Order Type Semiconductor Production Systems". In *2016 International Symposium on Semiconductor Manufacturing (ISSM)*, 12–13 December 2016, Tokyo, Japan.

Becker, L. T. and E. M. Gould. 2019. "Microsoft Power BI: Extending Excel to Manipulate, Analyze, and Visualize Diverse Data". *Serials Review* 45(3):184–188.

Beemsterboer, B., M. Land, and R. Teunter. 2016. "Hybrid MTO-MTS Production Planning: An Explorative Study". *European Journal of Operational Research* 248(2):453–461.

Chinosi, M. and A. Trombetta. 2012. "BPMN: An Introduction to the Standard". *Computer Standards & Interfaces* 34(1):124–134.

Forstner, L. and L. Monch. 2013. "A Heuristic to Support Make-to-Stock, Assemble-to-Order, and Make-to-Order Decisions in Semiconductor Supply Chains". In *2013 Winter Simulations Conference (WSC)*, 3696–3706 https://doi.org/10.1109/WSC.2013.6721730.

Ghasemi, A., A. Ashoori, and C. Heavey. 2021. "Evolutionary Learning Based Simulation Optimization for Stochastic Job Shop Scheduling Problems". *Applied Soft Computing* 106:107309.

Ghasemi, A., F. Farajzadeh, C. Heavey, J. Fowler, and C. T. Papadopoulos. 2024. "Simulation Optimization Applied to Production Scheduling in the Era of Industry 4.0: A Review and Future Roadmap". *Journal of Industrial Information Integration* 39:100599.

Ghasemi, A., C. Heavey, and K. E. Kabak. 2018. "Implementing a New Genetic Algorithm to Solve the Capacity Allocation Problem in the Photolithography Area". In *2018 Winter Simulation Conference (WSC)*, 3696–3707 https://doi.org/10.1109/WSC.2018.8632204.

Ghasemi, A., C. Heavey, and G. Laipple. 2018. "A Review of Simulation-Optimization Methods with Applications to Semiconductor Operational Problems". In *2018 Winter Simulation Conference (WSC)*, 3672–3683 https://doi.org/10.1109/WSC.2018.8632486.

Ivanov, D. and A. Dolgui. 2021. "A Digital Supply Chain Twin for Managing the Disruption Risks and Resilience in the Era of Industry 4.0". *Production Planning & Control* 32(9):775–788.

Kumar, R. and K. Singh. 2023. “High Utility Itemsets Mining from Transactional Databases: a Survey”. *Applied Intelligence* 53(22):27655–27703.

Lugaresi, G., Z. Jemai, and E. Sahin. 2024. “Digital Twins for Supply Chains: Main Functions, Existing Applications, and Research Opportunities”. In *2024 Winter Simulation Conference (WSC)*, 2076–2087 https://doi.org/10.1109/WSC60868.2023.10407569.

Maheshwari, P., S. Kamble, A. Belhadi, M. Venkatesh, and M. Z. Abedin. 2023. “Digital Twin-Driven Real-Time Planning, Monitoring, and Controlling in Food Supply Chains”. *Technological Forecasting and Social Change* 195:122799.

Mönch, L., R. Uzsoy, and J. W. Fowler. 2017. “A Survey of Semiconductor Supply Chain Models Part I: Semiconductor Supply Chains, Strategic Network Design, and Supply Chain Simulation”. *International Journal of Production Research* 56(13):4524–4545.

Park, K. T., Y. H. Son, and S. D. Noh. 2021. “The Architectural Framework of a Cyber Physical Logistics System for Digital-Twin-Based Supply Chain Control”. *International Journal of Production Research* 59(19):5721–5742.

Peeters, K. and H. Van Ooijen. 2020. “Hybrid Make-to-Stock and Make-to-Order Systems: a Taxonomic Review”. *International Journal of Production Research* 58(15):4659–4688.

San, O. 2021. “The Digital Twin Revolution”. *Nature Computational Science* 1(5):307–308.

Wagner, R., B. Schleich, B. Haefner, A. Kuhnle, S. Wartzack, and G. Lanza. 2019. “Challenges and Potentials of Digital Twins and Industry 4.0 in Product Design and Production for High Performance Products”. *Procedia CIRP* 84:88–93.

AUTHOR BIOGRAPHIES

AMIR GHASEMI is a Postdoctoral Research Fellow in the Institute of Applied Informatics and Formal Description Methods (AIFB) of the Karlsruhe Institute of Technology (KIT). His research interests include designing Simulation, Optimization, and Machine Learning-based Smart Agents in order to replace and/or support the human in decision making. His email address is: amir.ghasemi@kit.edu.

SANJA LAZAROVA-MOLNAR holds two full professorships, at the Karlsruhe Institute of Technology, and the University of Southern Denmark. Her research focuses on data-driven simulation, digital twins, and cyber-physical systems modeling for reliability and energy efficiency enhancement. Actively engaged in developing advanced methodologies, she leverages her expertise to optimize complex systems through data-driven simulation. She leads activities focused on digital twins and data-driven simulation in several European and national projects. Furthermore, Professor Lazarova-Molnar assumes leadership roles in IEEE and The Society for Modeling & Simulation International (SCS), contributing significantly to these professional organizations. She was also one of the Proceedings Editors for the Winter Simulation Conference in 2019 and 2020 and an associate editor of SIMULATION: Transactions of The Society for Modeling and Simulation International. Her email address is sanja.lazarova-molnar@kit.edu.

CATHAL HEAVEY is a Professor in the School of Engineering at the University of Limerick. He has published in the areas of queuing and simulation modeling. His research interests include simulation modeling of discrete-event systems; modeling and analysis of supply chains and manufacturing systems; process modeling; and decision support systems. His email address is cathal.heavey@ul.ie.