

A TOGAF-Driven Proactive Digital Twin for Production Planning and Resource Optimization

Zbigniew JUZON, Jarosław WIKAREK

Kielce University of Technology, Poland

Received: 02 March 2026

Accepted: 22 July 2026

Abstract

Abstract Modern production systems operate under high variability and frequent disruptions, reducing the effectiveness of classical static planning approaches. Although Digital Twin technologies enable real-time monitoring and simulation, many implementations lack embedded optimization mechanisms for prescriptive decision support. This paper proposes a TOGAF (The Open Group Architecture Framework)-driven Digital Twin framework for adaptive production planning, in which a Mixed-Integer Linear Programming (MILP) model is continuously re-optimized using real-time shop-floor data. The Digital Twin provides bidirectional synchronization between the physical system and its virtual decision model, enabling closed-loop adaptive control, dynamic resource reallocation, and proactive disturbance handling. The framework is structured within a TOGAF-aligned enterprise architecture that integrates business objectives, information flows, and optimization logic. The approach is validated through computational experiments based on an industrial case study, including static–dynamic comparison, scalability assessment, and robustness analysis under capacity disruptions. Results demonstrate a reduction in total completion time, an increase in average resource utilization, and preserved feasibility under severe disturbances, confirming the effectiveness and practical applicability of Digital Twin–based dynamic decision support.

Keywords

Digital Twin, resource allocation, production planning and scheduling, adaptive control, Industry 4.0.

Introduction

Modern manufacturing systems operate under conditions of increasing variability caused by product customization, short delivery cycles, fluctuating demand, and frequent disruptions such as machine breakdowns and labor shortages. In such circumstances, classic production planning methods based on static parameters and periodic rescheduling quickly lose their effectiveness. Schedules generated offline often become unfeasible during execution, leading to bottlenecks,

downtime, and reduced service levels. Consequently, production management increasingly requires decision-support mechanisms capable of continuously adapting plans to the current state of the shop floor. Recent advances in Industry 4.0 technologies have enabled extensive data acquisition from sensors, automation devices, and management solutions such as Enterprise Resource Planning (ERP) and Manufacturing Execution Systems (MESs). This development has stimulated interest in Digital Twin solutions, which provide synchronized virtual representations of physical production systems. Although Digital Twins improve monitoring and enable simulation or “what-if” analyses, many existing implementations remain descriptive. They support observation rather than automated decision-making and rarely integrate formal optimization methods that can generate prescriptive control actions. Concurrently, operations research has led to the development of numerous mathematical programming models for production planning and resource al-

Corresponding author: Zbigniew Juzon – Kielce University of Technology Faculty of Electrical Engineering, Automatic Control and Computer Science Department of Computer Science Applications, Al. 1000-lecia P.P. 7, 25-314 Kielce, Poland, phone: +, e-mail:

© 2026 The Author(s). This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

location. Mixed-Integer Linear Programming (MILP) formulas offer rigorous and repeatable decision support and can explicitly account for technological and performance constraints. However, such models are typically parameterized using static data defined at the beginning of the planning horizon. As a result, they respond poorly to real-time variability and often require manual rescheduling. These two streams – data-rich Digital Twins and optimization-based planning – have evolved largely independently. Digital Twin systems often lack embedded decision engines, and optimization models frequently operate without continuous synchronization with the physical system. Furthermore, the integration of these techniques within a coherent enterprise architecture that integrates strategic goals with operational control remains insufficiently addressed. In our previous article presented at the International Conference on Manufacturing Science (IMS 2025) [Wikarek et al. \(2025\)](#), we combined TOGAF-based organizational audits with a static MILP model to support production planning decisions. The study demonstrated that combining enterprise architecture analysis with mathematical optimization improved resource allocation; however, planning was performed as a one-time offline activity and did not incorporate real-time system feedback.

This paper extends this concept to dynamic and adaptive production control. The optimization model is embedded in a Digital Twin that continuously synchronizes with operational data and enables iterative re-optimization during execution. The solution is structured using a TOGAF-compliant enterprise architecture that formally connects business goals, applications, information flows, and technology infrastructure. This integration transforms the Digital Twin from a monitoring tool into a prescriptive decision-support mechanism capable of proactive rescheduling and resource reallocation.

The proposed framework is validated through computational experiments based on an industrial manufacturing case. Static and dynamic scenarios are compared using performance metrics such as lead time, resource utilization, backlog, and schedule stability, and additional scalability and resilience analyses are conducted. Results show that continuous, data-driven re-planning improves both performance and resilience to disruptions.

Literature review

Digital transformation in manufacturing increasingly focuses on improving flexibility, resilience, and transparency in highly variable production environ-

ments. Digital Twin (DT) technology has emerged as a central Industry 4.0 enabler, providing synchronized virtual representations of physical systems that support real-time monitoring, predictive analysis, and simulation-based decision support. Recent studies emphasize the evolution of Digital Twins from passive monitoring tools toward intelligent, data-driven decision systems ([Negri et al., 2021](#); [Cimino et al., 2024](#); [Rasheed et al., 2020](#)). In particular, integrated model-based Digital Twins have been shown to support joint maintenance and production planning decisions, improving operational robustness. In parallel, operations research has developed extensive mathematical programming models for production planning and resource allocation. MILP-based formulations provide rigorous and repeatable decision support but typically rely on static input parameters defined at the beginning of the planning horizon. As a result, their effectiveness decreases in environments characterized by dynamic disturbances and uncertainty. Although both Digital Twin systems and optimization-based planning models are well established, their integration into a coherent, closed-loop decision-making architecture remains limited. Most DT studies emphasize simulation and visualization rather than embedded optimization, while enterprise-architecture approaches rarely incorporate dynamic re-optimization mechanisms. Previous research work ([Sitek et al., 2014](#); [Sitek et al., 2018](#); [Bociewicz et al., 2022](#)) developed integrated optimization frameworks ([Wikarek et al., 2025](#)) and demonstrated the use of TOGAF-based enterprise architecture for static production planning. However, the combination of Digital Twin-enabled real-time data synchronization, MILP re-optimization, and enterprise-level architectural alignment has not been sufficiently validated through dynamic computational experiments. This research addresses that gap by embedding a MILP decision model within a Digital Twin framework structured according to TOGAF principles, enabling continuous parameter updates and proactive production control.

A TOGAF-Driven Proactive Digital Twin Approach

In the proposed architecture, TOGAF provides the structural and organizational framework, while the Digital Twin operationalizes this framework at the execution level. The MILP model is embedded as the decision and optimization core of the Digital Twin, whereas ERP/MES systems supply continuously updated production data. The approach proposed in this work systematically addresses the integration of Dig-

ital Twin technology with a formal decision-making model in the context of enterprise level production management. Its fundamental assumption is a departure from isolated, point-based IT solutions in favor of a coherent architecture that links the organization's strategic objectives with operational decisions made in real time. To this end, a corporate architecture-based approach, particularly aligned with TOGAF principles, is adopted as the structural foundation, allowing for clear definition of relationships between the business, application, information, and technology layers. At the business layer, the goals of the manufacturing enterprise, such as on-time order fulfillment, efficient resource utilization, and process resilience to disruptions must be considered. These goals serve as a reference point for the subsequent architecture layers and determine the requirements for the decision-support system. The application layer encompasses planning, scheduling, and optimization mechanisms. The central element of the proposed approach is Digital Twin technology, functioning as a dynamic bridge between the physical production system and its informational representation. The Digital Twin reflects the actual state of production resources, technological processes, and their interdependencies, remaining continuously synchronized with the physical system through data streams from automation and ERP/MES systems. Unlike static simulation models, the Digital Twin enables real-time updating of decision-model parameters, which is critical for making accurate decisions under conditions of variability and uncertainty. In the proposed approach, the Digital Twin does not serve merely as a monitoring tool but acts as an active component of the decision-making process. Advanced production control systems integrating Automated Guided Vehicle (AGV) and constraint-based scheduling approaches have been shown to improve operational efficiency in multimodal manufacturing environments (Kłosowski et al., 2018). Data generated by the Digital Twin are used for continuous calibration of the mathematical model, particularly regarding resource availability, operation times, and random events such as machine failures or material delays. As a result, the decision model transitions from a one-time planning tool to a mechanism for dynamic production control. Another important aspect of the proposed approach is the feedback loop between optimization results and the physical system. Decisions generated by the mathematical model based on real-time data provided by the Digital Twin can be fed into execution systems to adjust schedules or reconfigure resource allocation. This approach enables adaptive production control, where the system responds to disturbances not after they occur but based on early predictive signals. The proposed ap-

proach establishes a coherent environment integrating organizational, decision-making, and technological perspectives. Its structure allows the formal embedding of the mathematical model within the production facility architecture and its effective integration with Digital Twin technology. This serves as a foundation for further work on the mathematical formalization of the model and its verification through computational experiments, which are presented in subsequent sections. Model parameters, particularly resource availability and operation times, can change over successive planning periods based on updates from the Digital Twin system. While the reported operational improvements are directly obtained through Digital Twin-enabled iterative MILP optimization, the role of TOGAF is not to improve optimization performance by itself but to provide the enterprise architecture that enables systematic integration of heterogeneous production systems.

TOGAF ensures consistent alignment between business objectives, information flows, optimization services, and technological infrastructure. This architectural alignment facilitates interoperability between ERP, MES, IoT devices, and the Digital Twin, reducing integration complexity and improving maintainability and scalability of the decision-support system.

Without such an enterprise architecture, the Digital Twin and optimization model could still operate as isolated applications; however, extending the solution to additional production processes, organizational units, or industrial sites would require substantial redesign. Therefore, TOGAF contributes primarily by improving architectural consistency, modularity, interoperability, and long-term maintainability rather than directly affecting optimization KPIs.

Mathematical Model

To formally address the problem of planning/scheduling and allocating production resources in a multi-task environment, a deterministic mathematical model formulated for a discrete time horizon was adopted, based on the model presented in Wikarek et al. (2025). This model provides the decision-making basis for the proposed approach and enables quantitative analysis of the impact of production system parameter variability on the quality of generated schedules. Its structure enables integration with Digital Twin technology through dynamic updating of selected input parameters. Table 1 presents the indexes, parameters and decision variables of the model, while Table 2 presents a description of the model's constraints.

Table 1
Parameters, decision variables and indexes of the mathematical model

<i>Symbol</i>	<i>Description</i>
<i>Indexes</i>	
t	index of the planning period t , (T – planning horizon, $t \in (1..T)$)
i, j	product indices (I – set of all products, $i, j \in I$)
w	machine index (W – set of all machines, $w \in W$)
a	additional resource/worker index (a – set of all additional resources, $a \in A$)
<i>Parameters</i>	
$q_{ai,t}$	order quantity for product i in period t
pri	Determines if the order for the product must be fulfilled ($pri = 0$ “yes”; if $pri = 1$ “no”)
sui,j	Specifies how much product j is required to produce product i . If, $sui,j = 0$ product j is not directly included in product i
tej	Time t required to produce product j or deliver a product externally from a supplier
ppw,i	Coefficient $ppw,i = 1$ indicates that product i can be produced on machine w
szw,a	$szw,a = 1$: Resource a can be used on machine w . $szw,a = 0$: Resource a cannot be used on machine w
sri	Initial inventory of product i
zaa	Cost of using resource r per unit of time
pzw,i	Coefficient determines the time need to produce product i on machine w
psw,t	Production capacity of machine w in period t
mrw	Maximum number of product types that can be produced in period on machine w
sqi	Maximum allowable inventory for product i
pai	Time required to implement the basics of product i
<i>Decision variables</i>	
Di,t	Production quantity of product i in period t (Integer type)
Zi,w,a,t	If product i is produced in period t on machine w using resource a , then; $Zi,w,a,t = 1$ otherwise $Zi,w,a,t = 0$ (Binary type)
Ri,w,a,t	Production quantity of product i in period t on machine w , using resource a
Xi,t	Order quantity of product j that cannot be fulfilled in period t (Integer type)
Ui,t	Inventory level of product i at the end of period t
Fw,a,t	$Fw,a,t = 1$: Resource a in period t is assigned to machine w ; otherwise, $Fw,a,t = 0$ (Binary type)
Ha	If the additional resource r was used at least once then $Ha = 1$ otherwise $Ha = 0$ (Binary type)
Kt	If we work in the planning period t then $Kt = 1$ otherwise $Kt = 0$ (Binary type)

Table 1 continued on the next column

Table 1 continued from the previous column

<i>The quantities determined</i>	
K_R	Penalty for failure to complete orders – introduced so that the system would first plan production and only if it was not possible would it indicate that it was not possible to complete the order
W_D	Number of working days
C_A	Cost of using additional resources
W_T	Number of planning periods in which we work

Table 2
Description of model constraints

<i>Symbol</i>	<i>Description</i>
Ca_01	Planning production for customer orders; if this is not possible (e.g., due to insufficient capacity), variable B is set to indicate which products and in what quantities cannot be produced.
Ca_02a Ca_02b	Classic inventory balance equation (inventory at the end of the period equals inventory at the end of the previous period plus initiated production minus demand from fulfilling higher-level orders).
Ca_03	Planning production for a given product only from a period that allows enough time to complete the product and all its components (e.g., if the production cycle takes four periods, production cannot start in period 1, 2, or 3).
Ca_04	In a given planning period, production on a machine can only be planned within its capacity (CRP control).
Ca_05a Ca_05b	Correlation between variables R (production quantity) and Z (whether produced): if not produced, $Z = 0$, and R must be zero; if production is planned ($R > 0$), then $Z = 1$.
Ca_06a Ca_06b	Correlation between variable F (resource for production) and R (production quantity).
Ca_07	On a given machine in a planning period, only an allowed number of product types can be processed (e.g., due to setup time constraints).
Ca_08	In a given planning period, an additional resource can be assigned to only one machine.
Ca_09	The product inventory at the end of the period cannot exceed the allowed level.
Ca_10a Ca_10b	Correlation between variable K (work during the period) and F resource allocation during the period).
Ca_11a Ca_11b	Correlation between variable H (resource usage) and (resource allocation during the period).
D_01	Calculation of K_R (unfulfilled orders).
D_02	Determination of W_D (remaining tasks).
D_03	Determination of W_T (number of workdays)
D_04	Determination of C_A (additional costs).

The proposed model allows for answers to a number of decision-making questions. The most important of these are:

Q1:	<i>Are we able to fulfill all orders?</i> Feasibility is verified by checking whether all customer demands can be satisfied under constraints (Ca_01, Ca_02a–Ca_02b, Ca_05a–Ca_05b, Ca_09, D_01). A zero penalty value indicates full order fulfillment.
Q2:	<i>How to fulfill orders with the minimum number of workdays?</i> The production plan is optimized to minimize the number of working periods while penalizing unmet demand, subject to constraints (Ca_01, Ca_02a–Ca_02b, Ca_05a–Ca_05b, Ca_10a–Ca_10b, D_01, D_03).
Q3:	<i>How to fulfill orders with the lowest additional resource costs?</i> This objective minimizes the cost of employing additional resources while penalizing unmet demand, under constraints (Ca_01, Ca_04a–Ca_04b, Ca_05a–Ca_05b, Ca_08, Ca_10a–Ca_10b, Ca_11a–Ca_11b, D_01, D_04).
Q4:	<i>How to fulfill orders with the smallest number of additional resources?</i> Here, the number of additional resources is minimized while unmet demand is penalized, subject to constraints (Ca_01, Ca_04a–Ca_04b, Ca_05a–Ca_05b, Ca_08, Ca_10a–Ca_10b, Ca_11a–Ca_11b, D_01).
Q5:	<i>How to fulfill orders in the smallest number of periods?</i> This objective seeks the shortest completion horizon while penalizing unmet demand, under constraints (Ca_01, Ca_03, Ca_04a–Ca_04b, Ca_06a–Ca_06b, D_03).

Digital Twin Integration

To integrate Digital Twin technology with the mathematical model, it was necessary to present it in an implementation form. The implementation model was developed in AMPL (AMPL, 2026) and is presented in Appendix A (model.mod). The implementation model is a central element of the proposed approach to adaptive production system control. In addition to reactive rescheduling, the Digital Twin enables proactive decision-making through:

- early warning indicators based on resource utilization trends,
- evaluation of “what-if” scenarios and stress tests,
- prediction of future bottlenecks and capacity shortages,
- proactive rescheduling before disruptions occur.

Its core purpose is to ensure continuous, bidirectional information exchange between the physical sys-

tem and its decision-making representation, enabling dynamic adjustment of planning decisions to current production conditions. Unlike classical solutions, where the decision model relies on historical data or static assumptions, the use of a Digital Twin allows for real-time reflection of the actual state of the system. The Digital Twin functions as a digital intermediary layer between execution systems and the mathematical model. Data collected from automation systems, sensors, and MES-class systems reflect current production resource availability, actual operation times, workstation loads, and the occurrence of random events such as failures or downtime. This information is processed and aggregated in a manner that allows it to be directly used as input parameters for the mathematical model. A critical aspect of integration is the continuous parameter update mechanism of the mathematical model. Through synchronization with the Digital Twin, it is possible to account for changes in the production system almost in real time. This applies not only to short-term changes, such as temporary resource unavailability, but also to systemic changes resulting from production load trends or technical resource degradation. As a result, the mathematical model remains up to date and capable of generating decisions appropriate to the actual operational situation. Integration with the Digital Twin also enables predictive functionality. Based on historical data and real-time observations, the Digital Twin can identify patterns indicating potential disruptions in future planning periods. This information can be used to modify the parameters of the mathematical model before actual problems occur, supporting proactive rather than purely reactive decision-making. Another important aspect of integration is the feedback loop between the outputs of the mathematical model and the physical system. Decisions generated through optimization, such as schedule adjustments or resource reallocation, can be transmitted to production management systems for implementation. The effects of these decisions are then recorded again by the Digital Twin, closing the information loop and allowing continuous evaluation of the effectiveness of implemented actions. The system functions as a closed-loop adaptive control system with predictive feedback, continuously observing the process, predicting potential deviations, optimizing parameters, executing actions, and learning from outcomes (Abdul et al., 2025). The following diagram (Fig. 1) illustrates the integration of the Digital Twin into the production system, showing both the data flow from sensors to execution and the closed-loop adaptive control cycle that enables continuous observation, prediction, optimization, action, and learning.

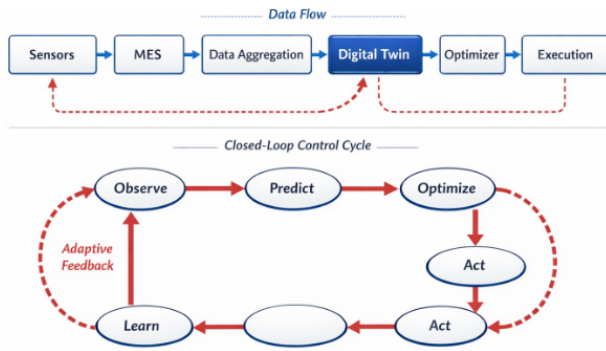


Fig. 1. Digital Twin Integration

The use of a Digital Twin as an integral part of the decision-making system represents a significant shift in the production management paradigm (Cimino et al., 2024). The mathematical model ceases to function merely as a tool for planning within a fixed time horizon and becomes part of a dynamic control mechanism capable of adapting to variability and uncertainty. This approach lays the foundation for quantitative assessment of the benefits resulting from the integration of the Digital Twin with the mathematical model, which is the focus of the computational experiment presented in the following section of this work.

Case Study Description

The considered production system, related to kitchen furniture manufacturing and described in the source article by Wikarek et al. (2025), executes a set of orders $\mathbf{P}$, each of which consists of a sequence of operations requiring the use of specific resources. The production resources form the set $\mathbf{R}$ and the planning process is conducted in discrete time periods $\mathbf{T}$. This scenario represents a classical production planning approach consistent with traditional TOGAF-based audits and static optimization models. The computational experiment is based on a reference production scenario. Monthly order volumes reflecting seasonal demand variability were adopted from Wikarek et al. (2025) and are used as baseline input data for the Digital Twin-based planning model (Table 3).

The monthly order volumes presented in Table 3 allow determination of the total demand for technological resources. Each order requires a specific number of hours for each resource, as shown in Table 4.

By multiplying the number of orders in a given month (Table 2) by the resource profiles for a single order (Table 3), the monthly resource demand I obtained. Further analysis and modeling will be conducted solely based on the data in Table 3, taking into account the scale of order volumes.

Table 3
Input Data (orders)

Month	Order Volume	Cumulative Orders
Jan	8	8
Feb	17	25
Mar	15	40
Apr	11	51
May	16	67
Jun	15	82
Jul	25	107

Table 4
Input Data (monthly resource demand)

Orders	Resources [h]			Total Hours [h]
	1	2	3	
P1	120	80	60	260
P2	90	110	70	270
P3	100	60	90	250

Computational Experiments

The computational experiment was conducted to empirically verify the effectiveness of the integrated approach, in which the mathematical model for production resource allocation is dynamically updated with data provided for initial availability of production resources. The optimization model was implemented using AMPL (A Mathematical Programming Language) and solved with Gurobi, allowing efficient handling of large-scale resource allocation problems (AMPL, 2026; Gurobi, 2026). AMPL is a high-level modeling language for describing and solving large-scale optimization problems. It allows users to define variables, constraints, and objective functions in a clear algebraic form, independent of the specific solver used. AMPL supports linear, nonlinear, integer, and stochastic optimization models and can interface with many solvers. Gurobi is a commercial state-of-the-art optimization solver designed for mathematical programming problems such as linear programming (LP), mixed-integer programming (MIP), and quadratic programming (QP). It is known for high performance, advanced presolve techniques, and parallel computation capabilities. Gurobi can be used through various programming languages and modeling systems, including AMPL. The implementation method is shown in Fig. 2.

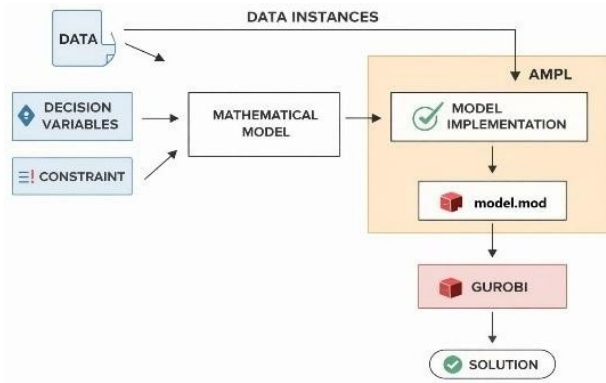


Fig. 2. Implementation method of the mathematical model

The case study represents an extension of the scenario described in the analyzed source article and pertains to a kitchen furniture production environment, as discussed by Wikarek et al. (2025), executing multiple parallel tasks under limited technical and temporal resources. The experiment was designed to enable a quantitative comparison of the decision-making system's behavior under static conditions versus dynamic conditions reflecting the actual operating environment of the production facility.

Experiment Scenarios

To isolate the impact of Digital Twin-based parameter updates, two computational scenarios were analyzed using the same mathematical model and objective structure.

Scenario 1: Static Planning

In the static scenario, all model parameters including resource availability $a_{r,t}$ and operation processing times $\mathbf{prp}, \mathbf{r}$ remain unchanged throughout the entire planning horizon. The model is solved once at the beginning of the horizon, and no adjustments are made in response to disturbances. This scenario represents a classical production planning approach consistent with traditional static optimization models as well as formal mathematical modeling, where the execution layer is implemented using mathematical programming techniques such as MP, CLP, or MILP. The process flow of the static planning scenario is illustrated in Fig. 3.

Scenario 2: Dynamic Planning (Digital Twin)

In the dynamic scenario, the physical production system generates operational data and disturbance signals that are transmitted to the decision support

subsystem and used to update the Digital Twin. Within the decision subsystem, based on the TOGAF architecture and MP/CLP/MILP models, the current process state is analyzed, alternative operational scenarios are simulated, and optimal control strategies are determined. Based on these analyses, the Digital Twin produces decision outputs in the form of operational recommendations and control setpoints. These decisions are then implemented in the physical plant, inducing changes in the process behavior and closing the feedback loop. The operation scheme of the Digital Twin is illustrated in Fig. 4.

To further formalize the proposed Digital Twin based decision-support architecture and address the enterprise-architecture perspective, a simplified ArchiMate 3.2 layered view derived from Fig. 5 is presented in Fig. 4.

Explicitly relating the proposed Digital Twin architecture to TOGAF principles (Table 5) provides an interpretation of the main components shown (Fig. 5) mapped onto the corresponding TOGAF architecture layers.

The sensitivity of the system to resource failures is illustrated in Fig. 1, which presents the time-dependent availability of Resource 2. While the static approach is unable to adapt to this disruption, the dynamic approach maintains feasibility and efficiency through adaptive rescheduling.

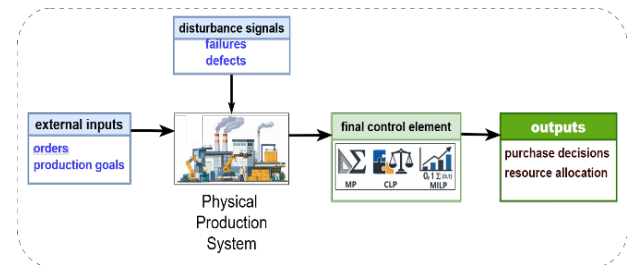


Fig. 3. Static Planning

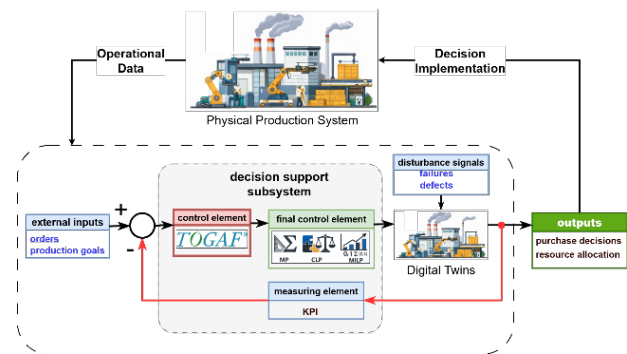


Fig. 4. Dynamic Production Control using Digital Twin

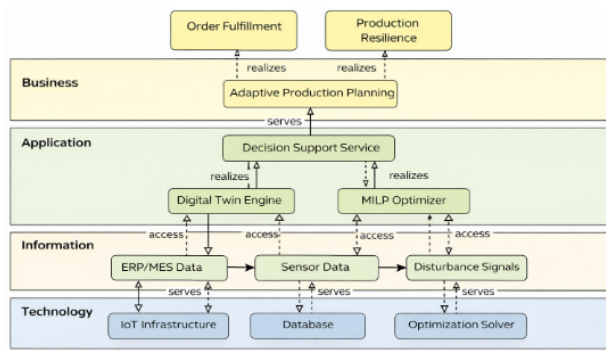


Fig. 5. Simplified ArchiMate 3.2 layered view of the Digital Twin based decision- support architecture derived from Fig. 3

Table 5

TOGAF layer interpretation of the Digital Twin architecture presented in Fig. 2

Layer	Elements (as shown in Fig. 2)
Business	Order fulfillment, production efficiency, system resilience
Application	MILP optimizer, Digital Twin engine, decision-support logic
Information	ERP data, sensor streams, KPIs, disturbance signals
Technology	IoT infrastructure, databases, optimization solvers

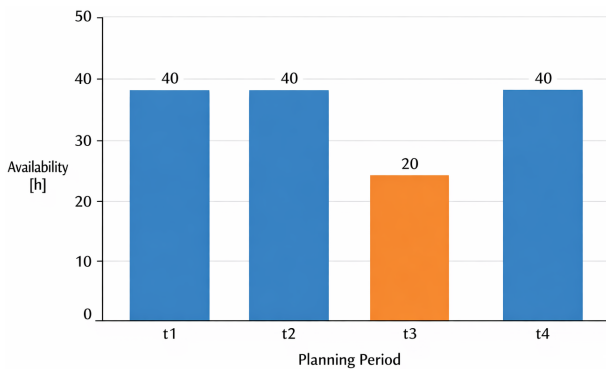


Fig. 6. Time-dependent Availability of Resource 2

At each planning period, the mathematical model is re-evaluated using the updated parameters, enabling adaptive reassignment of resources and rescheduling of operations.

Results

The mathematical model presented in Table 1 aims to optimize key aspects of the plant's operations, such as resource allocation, inventory management, and

order fulfillment. The model ensures that orders are executed in accordance with machine processing capacities, resource availability, and time constraints while simultaneously controlling inventory levels. It accounts for interdependencies between production and resource allocation, thereby improving operational efficiency. Furthermore, the model enables order acceptance only when sufficient production capacity is available, minimizing the risk of penalties for unfulfilled demand. Implementing this model enhances efficiency, flexibility, and scalability, allowing the plant to adapt to changing market conditions. Fig. 7 illustrates a production schedule in which workloads are assigned to individual machines across successive planning periods. For example, at station 9 in period 4, 288 units were produced, while in period 5 production increased to 625 units. The workload peaks observed in the static scenario originate from the model's inability to react to temporary capacity losses, which causes task accumulation in subsequent periods.

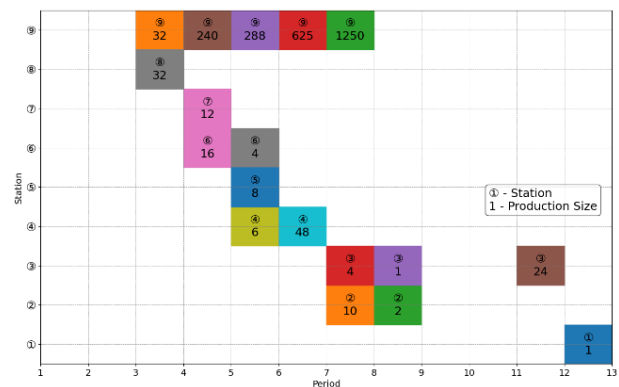


Fig. 7. Production Schedule by Station and Period

In contrast, the Digital Twin based dynamic approach proactively redistributes production when early warning signals of disturbances are detected, resulting in smoother utilization profiles and reduced idle periods. From an operational perspective, this translates into a lower risk of bottlenecks and overtime as well as improved shop-floor stability. Fig. 8 presents an alternative production schedule characterized by pronounced peaks and downtime periods. For instance, at station 9 in period 3, 240 units were produced, followed by a sharp increase to 1250 units in period 4. At station 6 in period 5, only 1 unit was completed, which rose to 16 units in period 6. Similarly, station 2 produced 25 units in period 6, indicating a temporary concentration of workload during that phase.

Comparing Fig. 7 and Fig. 8, it can be observed that the first plan ensures a more balanced distribution of production over time and across stations. In Fig. 6, pro-

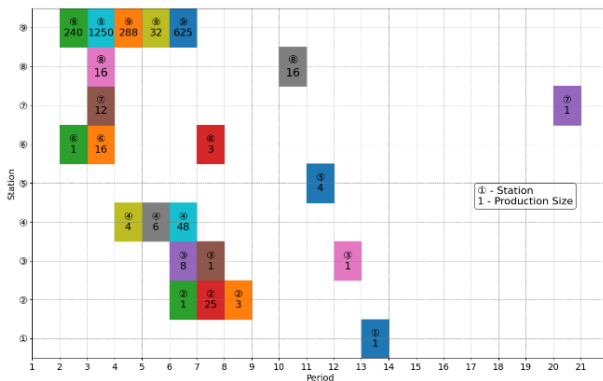


Fig. 8. Production Plan with Peaks and Downtime

duction at station 9 gradually increases from 288 units in period 4 to 625 units in period 5, thereby avoiding sudden overloads. In contrast, Fig. 7 exhibits extreme peaks, with production at the same station reaching 1250 units within a single period, which may result in machine overloading. Additionally, extended idle intervals can be observed at stations 6 and 7, indicating underutilization of resources, while other stations experience very high intensity, such as 48 units processed at station 4 in one period. Overall, the production plan depicted in Fig. 6 can be regarded as more optimal, as it promotes balanced resource utilization, reduces the likelihood of bottlenecks and idle periods, and ensures a more stable workflow. The aggregated performance indicators are summarized in Table 6.

Table 6

Performance comparison of static and dynamic planning

KPI	Static	Dynamic	Δ
Total Completion Time [h]	420	365	-13.1%
Resource Utilization [%]	72	84	+16.7%
Idle Time [h]	110	45	-59%
Backlog volume	18	3	-83%
Service level [%]	92	99	+7%
Schedule variability	high	low	—

The results presented in Table 6 clearly demonstrate the superiority of the Digital Twin-driven dynamic planning approach over the static planning scenario. In the static case, where model parameters remain fixed throughout the planning horizon, the total order completion time amounts to 420 h, and average resource utilization reaches 72%. By contrast, the dynamic scenario incorporates real-time updates of resource availability and processing times provided by

the Digital Twin. This leads to a reduction in total completion time to 365 h, corresponding to a decrease of approximately 13.1%, and an increase in average utilization to 84%. These improvements stem from the model's ability to continuously reassign resources and reschedule operations in response to disturbances such as temporary machine failures.

Scalability Analysis

To evaluate the computational scalability of the proposed MILP-based Digital Twin framework, additional experiments were conducted for progressively larger problem instances. In addition to the baseline case study, two expanded instances were generated by proportionally increasing the number of products (P), machines (M), and planning periods (T). The medium-scale instance doubled the size of these sets, while the large-scale instance tripled them. What is important and worth emphasizing is that, the model structure, objective function, and constraint system remained unchanged to ensure methodological consistency. All instances were solved using the same AMPL-Gurobi (Fig. 2) configuration and identical solver settings, allowing a fair comparison of computational performance. Scalability was assessed using three key indicators:

- number of decision variables,
- number of constraints,
- CPU computation time.

The results demonstrate a polynomial growth pattern in both model size and computation time. While the number of variables and constraints increases substantially with instance size, solution times remain within acceptable limits for medium-scale industrial environments. Even the largest instance ($\times 3$) was solved in under 100 seconds, confirming the practical tractability of the approach (Tab. 7). These findings indicate that embedding the MILP optimizer within a Digital Twin architecture does not compromise computational

Table 7
Scalability results

Inst.	P	M	T	Var.	Constr.	CPU
Baseline	17	9	10	12,450	21,300	3.1
Medium ($\times 2$)	34	18	20	45,820	79,600	18.4
Large ($\times 3$)	51	27	30	118,900	212,500	96.7

Abbreviations: *Inst.* – Instance size; *P* – Products; *M* – Machines; *T* – Periods; *Var.* – Decision variables; *Constr.* – Constraints; *CPU* – Computation time (seconds).

feasibility. The framework remains applicable to realistic production systems of moderate scale, which supports its practical deployment potential.

Sensitivity and Robustness Analysis

To assess the robustness of the proposed decision-support framework, a sensitivity analysis was conducted with respect to disruptions in production capacity and demand levels. The experiments considered:

- temporary capacity reductions of 20% and 40%,
- demand increases of 10% and 30% relative to baseline values,
- short-term unavailability of selected resources for one or two planning periods.

Two planning paradigms were compared:

- **Static scenario** – model parameters remain fixed throughout the planning horizon.
- **Dynamic Digital Twin scenario** – disturbed parameters are updated online and the MILP model is re-optimized at each planning period.

System performance was evaluated using:

- Total Completion Time (TCT),
- Resource Utilization (RU),
- Backlog volume (BL),
- Idle Time (IT).

The results (Tab. 8) clearly demonstrate that the static approach rapidly loses feasibility under severe disturbances. At a 40% capacity reduction, the static model becomes infeasible, as it cannot adjust to reduced availability. In contrast, the Digital Twin-based dynamic framework preserves feasibility by continuously reallocating resources and rescheduling operations.

Table 8
Robustness against capacity disruptions

CR [%]	S	TCT [h]	RU [%]	BL [jobs]	IT [h]
0%	Static	420	72	18	110
0%	Dynamic	365	84	3	45
20%	Static	455	66	31	140
20%	Dynamic	392	79	7	70
40%	Static	Infeasible	–	–	–
40%	Dynamic	435	71	15	105

Even under moderate disruptions (20% capacity loss), the dynamic approach achieves:

- shorter completion times,
- significantly lower backlog,
- improved utilization levels.

These results confirm that continuous parameter synchronization and iterative re-optimization substantially increase operational resilience. The Digital Twin architecture transforms the MILP model from a one-time planning tool into an adaptive control mechanism capable of maintaining performance under uncertainty. The presented validation is based on a representative industrial case study from the furniture manufacturing sector combined with computational experiments under different operating conditions. Although the obtained results demonstrate the feasibility and effectiveness of the proposed framework, they should not be interpreted as universally applicable quantitative improvements. The proposed architecture is intended as a generic integration framework rather than a domain-specific solution. Because the optimization model, Digital Twin synchronization mechanism, and TOGAF architecture are independent of a particular manufacturing process, the approach can be transferred to other discrete manufacturing environments after adapting the corresponding production constraints and resource models. Future validation should include multiple industrial case studies from different sectors such as automotive, electronics, and process manufacturing, together with pilot industrial implementations. The proposed framework differs from existing adaptive rescheduling solutions by combining three complementary elements within a single architecture: continuous Digital Twin synchronization, formal MILP optimization, and enterprise-wide integration based on TOGAF principles. While iterative re-optimization improves operational performance, TOGAF provides the architectural foundation that ensures consistent information flow, interoperability between ERP/MES systems and optimization services, and alignment between business objectives and production decisions. Therefore, the contribution of the proposed approach extends beyond adaptive rescheduling by providing an enterprise-level architecture for scalable and maintainable Digital Twin deployment.

Conclusions

The implemented mathematical model enables the optimization of resource allocation, inventory management, and order fulfillment, taking into account machine capacities, resource availability, and time constraints. Integrating the model with the TOGAF architecture and a Digital Twin allows for real-time parameter updates, significantly enhancing the efficiency and resilience of the production system. It should be emphasized that the observed improvements in operational KPIs result from the Digital Twin-based adap-

tive optimization, whereas TOGAF provides the architectural framework that enables systematic integration, interoperability, and scalability of the proposed decision-support solution. In the dynamic scenario utilizing the Digital Twin, the total completion time (TCT) decreased from 420 h to 365 h, while average resource utilization (RU) increased from 72% to 84% compared to the static scenario, as shown in Table 8. Dynamic parameter updates allow for early detection of bottlenecks, adaptive resource assignment, and the maintenance of a stable production flow, as illustrated in Tables 7 and 8. The results indicate that combining the TOGAF based model with a Digital Twin provides an effective tool for adaptive production control in highly variable and uncertain environments, linking enterprise-level strategic objectives with operational decision-making and improving the quality of real-time decisions. Scalability tests confirm that the framework remains computationally tractable for medium-scale industrial systems. Robustness experiments show that the approach maintains feasibility even when static planning becomes infeasible. From a managerial perspective, embedding an optimization-driven Digital Twin within a TOGAF enterprise architecture strengthens the alignment between strategic goals and operational execution. The framework transforms the Digital Twin from a descriptive monitoring tool into a prescriptive and proactive decision-support system. Future research may extend the model toward stochastic programming formulations, multi-objective optimization, and machine-learning-enhanced parameter forecasting, further increasing adaptability in highly uncertain production environments.

References

- Abdul R.P., Dallasega I., Kaushal A., Revolti H., & Medina Suni A. (2025). Digital Twin prototype to support factory and production planning. *IFAC PapersOnLine*, 59(10), 939–944. [10.1016/j.ifacol.2025.09.159](https://doi.org/10.1016/j.ifacol.2025.09.159)
- AMPL. (2026). Home AMPL. AMPL. Retrieved February 25, 2026, from <https://ampl.com>
- Błażewicz J. (2014). *Handbook on Scheduling*. Springer-Verlag Berlin and Heidelberg GmbH & Co.
- Bocewicz G., Sitek P., Wikarek J., & Nielsen I. (2022). A decision support model for handling customer orders in business chain. *Neurocomputing*, 482, 298–309. [10.1016/j.neucom.2021.06.099](https://doi.org/10.1016/j.neucom.2021.06.099)
- Cimino C., El Warraqi L., & Negri E. (2024). Digital twins based on integrated models: Supporting joint decisions on maintenance and production planning. *IFAC PapersOnLine*, 57-XX. [10.1016/j.ifacol.2024.08.044](https://doi.org/10.1016/j.ifacol.2024.08.044)
- Gurobi. (2026). Gurobi The Fastest Solver. Gurobi. Retrieved February 25, 2026, from <https://gurobi.com>
- Kłosowski G., Gola A., & Thibbotuwawa A. (2018). Computational intelligence in control of AGV multimodal systems. *IFAC PapersOnLine*, 51(11), 1421–1427. [10.1016/j.ifacol.2018.08.315](https://doi.org/10.1016/j.ifacol.2018.08.315)
- Negri E., Fumagalli L., & Macchi M. (2020). A review of the roles of Digital Twin in CPS-based production systems. In A. Crespo Márquez, M. Macchi, & A. Parlikad (Eds.), *Value Based and Intelligent Asset Management*. Springer, Cham. [10.1007/978-3-030-20704-5_13](https://doi.org/10.1007/978-3-030-20704-5_13)
- Rasheed A., San O., & Kvamsdal T. (2020). Digital Twin: Values, challenges and enablers from a modeling perspective. *IEEE Access*, 8, 21980–22012. [10.1109/ACCESS.2020.2970143](https://doi.org/10.1109/ACCESS.2020.2970143)
- Sitek P., & Wikarek J. (2014). A novel integrated approach to the modelling and solving of the Two-Echelon Capacitated Vehicle Routing Problem. *Production & Manufacturing Research*, 2(1), 326–340. [10.1080/21693277.2014.910716](https://doi.org/10.1080/21693277.2014.910716)
- Sitek P., & Wikarek J. (2018). A multi-level approach to ubiquitous modeling and solving constraints in combinatorial optimization problems in production and distribution. *Applied Intelligence*, 48, 1344–1367. [10.1007/s10489-017-1107-9](https://doi.org/10.1007/s10489-017-1107-9)
- Wikarek J., Juzoń Z., & Sitek P. (2025). Optimization of production process through TOGAF-based audits and mathematical programming models. *IFAC PapersOnLine*, 59(24), 30–35. [10.1016/j.ifacol.2025.11.836](https://doi.org/10.1016/j.ifacol.2025.11.836)
- Xu X. (2017). Machine Tool 4.0 for the new era of manufacturing. *The International Journal of Advanced Manufacturing Technology*, 92. [10.1007/s00170-017-0300-7](https://doi.org/10.1007/s00170-017-0300-7)

Appendix A.

model.mod

```

set ST
set SI
set SW
set SA
param pr{SI}
param sr{SI}
param te{SI}
param pa{SI}
param sq{SI}
param mr{SW}
param za{SA}
param qa{SI,ST}
param kr_z{SI,ST}
param su{SI,SI}
param sz{SW,SA}
param pp{SW,SI}
param pz{SW,SI}
param ps{SW,ST}
param sta
param TZ
var K{ST} >=0, binary
var D{SI,ST} >=0, integer
var X{SI,ST} >=0, integer
var U{SI,ST} >=0, integer
var F{SW,SA,ST} >=0, binary
var R{SI,SW,SA,ST} >=0, integer
var Z{SI,SW,SA,ST} >=0, binary
var H{SA} >=0, binary
var K_R
var W_D
var W_T
var C_A
subject to C_01 {j ∈ SI, t ∈ ST : t + te[j] < TZ} :
    D[j, t] = qa[j, t] - pr[j] · X[j, t] +
    ∑i ∈ SI (su[i, j] · ∑w ∈ SW, a ∈ SA R[i, w, a, t + te[j]]);
subject to C_02a {i ∈ SI, t ∈ ST : t ≥ 2} :
    U[i, t - 1] +
    ∑w ∈ SW, a ∈ SA (pp[w, i] · sz[w, a] · R[i, w, a, t]) =
    D[i, t] + U[i, t];
subject to C_02b {i ∈ SI, t ∈ ST : t = 1} :
    sr[i] + ∑w ∈ SW, a ∈ SA (pp[w, i] · sz[w, a] · R[i, w, a, t]) =
    D[i, t] + U[i, t];
subject to C_03 {i ∈ SI, w ∈ SW, a ∈ SA, t ∈ ST : t >
    pa[i]} : R[i, w, a, t] = 0;
subject to C_04 {w ∈ SW, t ∈ ST} :
    ∑i ∈ SI, a ∈ SA pz[w, i] · R[i, w, a, t] ≤ ps[w, t];
subject to C_05a {i ∈ SI, w ∈ SW, a ∈ SA, t ∈ ST :
    sz[w, a] ≠ 0} :
    Z[i, w, a, t] ≤ R[i, w, a, t];

```

Model Constraints Continuation on the next column

Model Constraints Continuation from the previous column

```

subject to C_05b {i ∈ SI, w ∈ SW, a ∈ SA, t ∈ ST :
    sz[w, a] ≠ 0} :
    sta · Z[i, w, a, t] ≥ R[i, w, a, t];
subject to C_06a {w ∈ SW, a ∈ SA, t ∈ ST : sz[w, a] ≠ 0} :
    F[w, a, t] ≤ ∑i ∈ SI R[i, w, a, t];
subject to C_06b {w ∈ SW, a ∈ SA, t ∈ ST : sz[w, a] ≠ 0} :
    sta · F[w, a, t] ≥ ∑i ∈ SI R[i, w, a, t];
subject to C_07 {w ∈ SW, t ∈ ST} :
    ∑i ∈ SI, a ∈ SA Z[i, w, a, t] ≤ mr[w];
subject to C_08 {a ∈ SA, t ∈ ST} :
    ∑w ∈ SW F[w, a, t] ≤ 1;
subject to C_09 {i ∈ SI, t ∈ ST} :
    U[i, t] ≤ sr[i];
subject to C_10a {t ∈ ST} :
    K[t] ≤ ∑w ∈ SW, a ∈ SA F[w, a, t];
subject to C_10b {t ∈ ST} :
    sta · K[t] ≥ ∑w ∈ SW, a ∈ SA F[w, a, t];
subject to C_11a {a ∈ SA} :
    H[a] ≤ ∑w ∈ SW, t ∈ ST F[w, a, t];
subject to C_11b {a ∈ SA} :
    sta · H[a] ≥ ∑w ∈ SW, t ∈ ST F[w, a, t];
subject to D_01:
    K_R = sta · ∑i ∈ SI, t ∈ ST X[i, t];
subject to D_02:
    W_D = ∑i ∈ SI, w ∈ SW, a ∈ SA, t ∈ ST Z[i, w, a, t];
subject to D_03:
    W_T = ∑t ∈ ST K[t];
subject to D_04:
    C_A = ∑w ∈ SW, a ∈ SA, t ∈ ST za[a] · F[w, a, t];

```

Model Data Configuration – Full Dataset

data;
set ST := 1 2 3 4 5 6 7 8 9 10;
set SI := P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17;
set SW := M01 M02 M03 M04 M05 M06 M07 M08 M09;
set SA := R01 R02 R03 R04 R05 R06 R07 R08 R09 R10;
param pr :=
P01 1 P06 0 P11 0 P16 0
P02 0 P07 0 P12 0 P17 0
P03 0 P08 0 P13 0
P04 0 P09 0 P14 0
P05 0 P10 0 P15 0
param sr :=
P01 0 P06 0 P11 0 P16 0
P02 0 P07 0 P12 0 P17 0
P03 0 P08 0 P13 0
P04 0 P09 0 P14 0
P05 0 P10 0 P15 0
param sq :=
P01 10 P06 300 P11 500 P16 100
P02 100 P07 50 P12 10000 P17 1000
P03 100 P08 50 P13 9000
P04 50 P09 50 P14 1000
P05 200 P10 300 P15 1000
param te :=
P01 1 P06 1 P11 1 P16 1
P02 1 P07 1 P12 1 P17 1
P03 1 P08 1 P13 1
P04 1 P09 1 P14 1
P05 1 P10 1 P15 1
param pa :=
P01 4 P06 1 P11 2 P16 1
P02 2 P07 1 P12 1 P17 1
P03 3 P08 2 P13 1
P04 3 P09 2 P14 1
P05 2 P10 1 P15 1

Model Data Configuration – Full Dataset (Continued on the next page)

Model Data Configuration – Full Dataset (Continued from the previous page)

param su :=																	
	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	P13	P14	P15	P16	P17
P01	0	8	4	2	25	0	0	0	0	0	0	0	0	0	0	0	0
P02	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
P03	0	0	0	0	0	0	1	0	0	8	12	0	0	0	0	0	0
P04	0	0	0	0	0	0	0	3	4	0	0	0	0	0	0	0	0
P05	0	0	0	0	0	0	0	0	0	0	0	50	25	0	0	0	0
P06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
P09	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
P10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P11	0	0	0	0	0	0	0	0	0	0	0	0	0	6	5	0	0
P12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

param pp :=																	
	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	P13	P14	P15	P16	P17
M01	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M02	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
M03	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
M04	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
M05	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
M06	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
M07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
M08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M09	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	1

param pz :=																	
	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	P13	P14	P15	P16	P17
M01	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M02	0	180	180	360	20	0	0	0	0	0	0	0	0	0	0	0	0
M03	0	180	180	360	20	0	0	0	0	0	0	0	0	0	0	0	0
M04	0	0	0	0	0	0	0	180	180	0	50	0	0	0	0	0	0
M05	0	0	0	0	0	0	0	180	180	0	50	0	0	0	0	0	0
M06	0	0	0	0	0	30	180	0	0	0	0	0	0	0	0	0	0
M07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
M08	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	50	0
M09	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	1

Model Data Configuration – Full Dataset (Continued on the next page)

Model Data Configuration – Full Dataset (Continued from the previous page)

param ps :=										
	1	2	3	4	5	6	7	8	9	10
M01	28800	28800	28800	28800	28800	28800	28800	28800	28800	28800
M02	28800	28800	28800	28800	28800	28800	28800	28800	28800	28800
M03	28800	28800	28800	28800	28800	28800	28800	28800	28800	28800
M04	28800	28800	28800	28800	28800	28800	28800	28800	28800	28800
M05	28800	28800	28800	28800	28800	28800	28800	28800	28800	28800
M06	28800	28800	28800	28800	28800	28800	28800	28800	28800	28800
M07	28800	28800	28800	28800	28800	28800	28800	28800	28800	28800
M08	28800	28800	28800	28800	28800	28800	28800	28800	28800	28800
M09	99999	99999	99999	99999	99999	99999	99999	99999	99999	99999

param mr :=										
M01	1	M04	2	M07	1					
M02	3	M05	2	M08	1					
M03	3	M06	1	M09	20					

param za :=										
R01	100	R04	100	R07	300	R10	900			
R02	200	R05	200	R08	900					
R03	300	R06	100	R09	1300					

param qa :=										
	1	2	3	4	5	6	7	8	9	10
P01	0	0	0	0	22	0	10	0	0	0
P02	0	0	0	0	0	0	0	0	0	0
P03	0	0	0	0	0	0	0	0	0	0
P04	0	0	0	0	0	0	0	0	0	0
P05	0	0	0	0	0	0	0	0	0	0
P06	0	0	0	0	0	0	0	0	0	0
P07	0	0	0	0	0	0	0	0	0	0
P08	0	0	0	0	0	0	0	0	0	0
P09	0	0	0	0	0	0	0	0	0	0
P10	0	0	0	0	0	0	0	0	0	0
P11	0	0	0	0	0	0	0	0	0	0
P12	0	0	0	0	0	0	0	0	0	0
P13	0	0	0	0	0	0	0	0	0	0
P14	0	0	0	0	0	0	0	0	0	0
P15	0	0	0	0	0	0	0	0	0	0
P16	0	0	0	0	0	0	0	0	0	0
P17	0	0	0	0	0	0	0	0	0	0

param TZ = 10;										
param sta = 10000;										