applied sciences

MDPI

Article

An Exact Continuous-Time Markov Chain Framework for Modeling and Performance Evaluation of Multi-Product Push–Pull Production Systems

Angelos Kourepis [1,*], Alexandros C. Diamantidis [2], Stelios Koukoumialos [3], Nikolaos Kladovasilakis [4,*] and Michael A. Madas [1]

[1] Department of Applied Informatics, University of Macedonia, 546 36 Thessaloniki, Greece; mmadas@uom.gr
[2] Department of Economics, Aristotle University of Thessaloniki, 541 24 Thessaloniki, Greece; adiama@econ.auth.gr
[3] Department of Business Administration, University of Thessaly, 415 00 Larissa, Greece; skoukoum@uth.gr
[4] Department of Supply Chain Management, International Hellenic University (IHU), 601 00 Katerini, Greece
* Correspondence: kourepis.angelos@gmail.com (A.K.); n.kladovasilakis@ihu.edu.gr (N.K.)

Abstract

Multi-product production systems require effective coordination among production, intermediate storage, downstream processing, and customer demand, particularly under stochastic operating conditions and finite capacity. Unlike existing analytical studies that mainly examine either single-product push–pull systems or multi-product manufacturing systems separately, this work develops an exact CTMC framework that jointly captures product variety, shared buffering, downstream parallelization, sequence-dependent setups, and machine unreliability. The system comprises an unreliable upstream machine with sequence-dependent setup changes, a finite intermediate buffer, a distribution center modeled as a pooled processing resource with (M) identical reliable channels, and dedicated finished-goods buffers serving product-specific demand. A high-dimensional continuous-time Markov chain is formulated, and a systematic algorithm is developed to construct the infinitesimal generator matrix and compute steady-state performance measures. Numerical experiments examine intermediate buffer capacity, downstream processing capacity, priority rules, and upstream machine reliability. Increasing buffer capacity from 0 to 20 increases total throughput from 0.5937 to 1.0988, whereas further expansion to 100 yields only 1.1606, while average work-in-process reaches 19.8524. Downstream capacity exhibits similar diminishing performance gains, while priority rules and machine reliability affect product-level and overall throughput. These findings highlight throughput–inventory trade-offs and demonstrate the framework's applicability for evaluating alternative configurations.

Keywords: multi-product production systems; push–pull systems; continuous-time Markov chains; performance evaluation; finite buffers; parallel machines

Check for updates

Academic Editor: Paolo Renna

Received: 8 August 2026
Revised: 7 September 2026
Accepted: 9 September 2026
Published: 10 September 2026

1. Introduction

Modern manufacturing and supply chain systems operate in increasingly complex environments characterized by product variety, fluctuating customer demand, limited production and storage capacities, and equipment unreliability. Under these conditions, the effective coordination of production, inventory, and material flows becomes essential for maintaining high operational performance. The increasing demand for product variety has further encouraged the adoption of flexible manufacturing systems that accommodate different product types using common production resources [1,2]. At the same time, the transition toward smart manufacturing and Industry 4.0 has reinforced the importance of data-driven decision-making, real-time monitoring, and analytical methods that support production system design and operational control [3,4]. Consequently, mathematical models that capture stochastic system dynamics and quantify the effects of alternative design and operating parameters constitute important decision-support tools for modern manufacturing systems.

Hybrid push–pull production systems provide an effective mechanism for managing production activities in response to external customer demand. In these systems, upstream operations generally follow push-based production principles, whereas downstream operations respond to actual demand through pull-based mechanisms [5,6]. The intermediate inventory separating these two operating regimes plays a critical role in coordinating material flows and limiting the propagation of production variability. The performance of such systems is influenced by several interacting factors, including buffer capacity, production and processing rates, machine reliability, downstream capacity, external demand, and production-control policies. Their simultaneous consideration becomes particularly challenging in multi-product environments, where different product types compete for shared production and storage resources and may be subject to distinct demand rates and processing priorities.

Analytical and stochastic modeling methods have therefore been extensively employed to evaluate manufacturing and production–inventory systems. Among these approaches, Markovian models provide a structured framework for representing systems in which production completions, machine failures and repairs, setup changes, and demand arrivals occur randomly over time. Fundamental contributions to the analysis of stochastic manufacturing systems established the applicability of Markov processes to unreliable production lines and finite-buffer systems [7–9]. More recently, Markovian approaches have continued to support the analysis of increasingly complex manufacturing configurations involving finite storage, parallel processing resources, multiple product types, and alternative operating policies [10–12].

Within the literature on multi-product manufacturing systems, several analytical approaches have been developed to address the additional complexity arising from shared resources and product interactions. Nemec [13] and Colledani et al. [14] employed decomposition-based approaches to estimate the performance of multi-product production lines, while Mastrangelo et al. [15] combined analytical and simulation models to evaluate the performance of reconfigurable manufacturing systems. Syrowicz [16] investigated multiple-part-type production lines with multiple failure modes, whereas Li et al. [17] developed an iterative analytical approach for manufacturing systems involving split and merge operations. Diamantidis et al. [18] analyzed a push–merge manufacturing configuration with unreliable machines and multiple product types using a Markovian framework. These studies demonstrated the importance of explicitly representing product interactions when evaluating manufacturing system performance.

Further developments have focused specifically on stochastic multi-product systems with finite buffers and product-dependent operating characteristics. Feng et al. [19] modeled a multi-product manufacturing system with sequence-dependent setup times and finite buffers as a continuous-time Markov chain and evaluated alternative scheduling policies. Kang et al. [20] subsequently investigated systems with arbitrary processing-time distributions using an embedded Markov chain and approximation techniques. These studies [19,20] focus primarily on the coordination and scheduling of multiple products within manufacturing lines. They do not consider the integrated push–pull structure examined in the present study, which incorporates product-specific finished-goods inventories and external demand, an unreliable shared upstream production resource, and pooled parallel downstream processing governed by a strict product-priority rule.

Park et al. [21] employed Markov chain analysis to estimate the throughput of a multi-product machining line in an industrial application, while Chen et al. [22] examined flexible production lines with dedicated buffers under a Bernoulli reliability model. Collectively, these contributions demonstrate the suitability of Markovian and related stochastic approaches for capturing the interactions among product variety, production resources, storage constraints, and operating policies.

Kourepis et al. [23] and Diamantidis et al. [24], in contrast, provide analytical foundations for push–pull systems involving intermediate storage and distribution operations. Kourepis et al. [23] consider multiple non-identical suppliers and retailers under supply disruptions, whereas Diamantidis et al. [24] examine a push–pull merge system with multiple suppliers and parallel distribution-center channels. Although these studies capture important push–pull, reliability, and parallel-processing characteristics, they do not jointly represent multiple product types sharing the same upstream production resource, sequence-dependent switching among product settings, product-dependent operational and failed modes, competition for pooled downstream-processing capacity, strict product priorities, and differentiated external demand.

Accordingly, the present study does not claim that finite buffers, multiple products, parallel processing, machine unreliability, priority rules, or push–pull control are individually new modeling elements. Its incremental methodological contribution lies in integrating these elements into a unified finite-state CTMC and providing a systematic, reproducible procedure for constructing the corresponding state space and sparse infinitesimal generator matrix. The proposed construction explicitly maps every admissible system state and each feasible production, setup, failure, repair, downstream-processing, and demand event to the appropriate generator-matrix element. This enables the direct steady-state evaluation of product-specific throughput, total throughput, and inventory performance under a common analytical framework, without state aggregation, decomposition, approximation, or simulation-based estimation.

The remainder of this paper is organized as follows. Section 2 presents the proposed analytical framework. Section 3 reports the numerical results and sensitivity analyses. Section 4 discusses the main findings and their managerial implications. Finally, Section 5 concludes the paper and outlines future research directions.

2. Methods

2.1. Methodological Framework

The methodology adopted in this study follows an approach for the analytical modeling and performance evaluation of the proposed multi-product push–pull production system. Initially, the system configuration, operating characteristics, and modeling are defined. Based on the CTMC formulation, the infinitesimal generator matrix is systematically constructed, and the steady-state probabilities are obtained, thereby enabling the calculation of the selected performance measures. Finally, a parametric analysis is conducted to explore the effects of key structure and operational parameters. The overall methodological procedure is summarized in Figure 1.

Figure 1. Methodological framework for the CTMC-based modeling and performance evaluation of the proposed multi-product push–pull production system.

2.2. Production System Description

Following the approach illustrated in Figure 1, the production system considered in this study is defined as a two-stage multi-product push–pull configuration comprising four interacting components: (i) the upstream disruption-prone machine, whose operational state determines the product type being manufactured; (ii) the intermediate buffer with finite capacity, which accumulates semi-finished items; (iii) the distribution center (DC) with M pooled parallel channels; and (iv) three finished-goods buffers, each dedicated to a specific product type. The (M) downstream channels are not modeled as independent servers assigned to different items. Instead, their processing capacities are pooled and collectively allocated to one eligible product type at a time. Consequently, the distribution center performs a consolidated processing operation with an aggregate rate that increases linearly with M. The overall objective is to describe the stochastic behavior of production, failures, repairs, material flows (exponentially distributed), and external customer demand (Poisson-Distributed) by formulating a continuous-time Markov chain (CTMC) with a discrete state space [25]. In the following model, at most one event can occur during an infinitesimal interval (dt).

2.3. Continuous-Time Markov Chain Formulation

2.3.1. State Description

The system is modeled as a CTMC with a discrete state space. At any time t, the state of the system is described by the vector:

$$S(t) = (Q(t), n_A(t), n_B(t), n_C(t), F_A(t), F_B(t), F_C(t)) \tag{1}$$

- $Q(t) \in \{A, B, C, D_A, D_B, D_C\}$ denotes the state of the upstream machine, with states A, B, C, corresponding to the production of the respective product types. In contrast, D_i denotes the failure states of the machine when producing product type i.
- $n_i(t)$ denotes the number of semi-finished items of product type $i \in \{A, B, C\}$ stored in the intermediate buffer and in the one additional slot of the machine, when it is blocked.
- $F_i(t)$ denotes the inventory level of finished goods of product type $i \in \{A, B, C\}$ in the corresponding finished-goods buffer.
- No additional state variable is introduced for the occupancy or assignment of the individual downstream channels. This omission is deliberate because the distribution center is modeled as an aggregate pooled processing resource rather than as a conventional multi-server queue. The pooled resource is collectively assigned to one eligible product type at a time, and the selected item remains included in $n_i(t)$ until the corresponding downstream completion event occurs. The allocation is re-evaluated after every state transition. Given the current values of $n_i(t)$ and $F_i(t)$, together with the strict priority conditions, the applicable downstream completion rate and successor state are uniquely determined. Moreover, because the downstream completion time is exponentially distributed, no residual processing-time variable is required. The state vector in Equation (1) is therefore sufficient to preserve the Markov property for the aggregate pooled-capacity model considered in this study.

The state variables are subject to the constraints:

$$0 \leq \sum_{i \in \{A,B,C\}} n_i(t) \leq B_1 + 1 \tag{2}$$

and

$$0 \leq F_i(t) \leq F_i^{max}(t),\ i \in \{A, B, C\} \tag{3}$$

where B_1 denotes the total capacity of the intermediate buffer and F_i^{max} the capacity of the finished-goods buffer for product type i.

2.3.2. Upstream Disruption-Prone Machine Dynamics

The upstream machine has an infinite supply of workpieces to be processed and operates under a blocking-after-service policy. The total physical storage capacity associated with the upstream machine is denoted by (C). It consists of the intermediate buffer of capacity (B_1) and one additional slot at the machine itself. Hence,

$$C = B_1 + 1 \tag{4}$$

Production

When the upstream machine is in productive state $i \in \{A, B, C\}$ and the physical storage capacity (C) is not full, i.e.,

$$\sum_{i \in \{A,B,C\}} n_i(t) < B_1 + 1 \tag{5}$$

a service completion generates one semi-finished item of type $i \in \{A, B, C\}$, corresponding to the CTMC transition:

$$S_t(Q = i, n_i, F_i) \to S_{t+dt}(Q = i, n_i + 1, F_i) \tag{6}$$

with rate μ_i.

If $\sum_{i \in \{A,B,C\}} n_i(t) = B_1 + 1$, the machine becomes blocked after service completion. No production, setup changes, or failures take place while the machine is blocked; they may occur only once space becomes available in the intermediate buffer.

Setup changes

While in a productive state $i \in \{A, B, C\}$, the machine may switch its production setting to another product type $j \in \{A, B, C\}$, $(j \neq i)$. This corresponds to the transition:

$$S_t(Q = i, n_i, F_i) \to S_{t+dt}(Q = j, n_i, F_i) \tag{7}$$

with rate α_{ij}.

In the general formulation, α_{ij} depends on both the current production setting i and the destination product type j. Therefore, the off-diagonal setup-transition rates may differ across production sequences and directions, allowing the proposed framework to represent sequence-dependent setup changes. The diagonal elements satisfy $\alpha_{ii} = 0$ since no setup transition occurs when the production setting remains unchanged.

Failures and Repairs

Machine failures may occur only when the machine is operational $i \in \{A, B, C\}$ and not blocked (Equation (5)). A failure while producing product $i \in \{A, B, C\}$ corresponds to:

$$S_t(Q = i, n_i, F_i) \to S_{t+dt}(Q = D_i, n_i, F_i) \tag{8}$$

with rate p_i.

When the machine is in the failure state D_i, $i \in \{A, B, C\}$, it undergoes repair and returns to the productive state it occupied immediately before failure:

$$S_t(Q = D_i, n_i, F_i) \to S_{t+dt}(Q = i, n_i, F_i) \tag{9}$$

with rate r_i.

2.3.3. Flow Dynamics at the Intermediate Buffer

Semi-finished items are stored in the intermediate buffer, which is shared across all product types and has a finite total capacity (B_1). No dedicated capacity is assigned to specific product types; instead, the buffer holds any feasible combination of items, on condition that the total number does not exceed the buffer's overall storage capacity (B_1).

Hence, the vectors (n_i) capture both the composition and the congestion level of the intermediate buffer and directly affect the blocking behavior of the upstream machine as well as the availability of jobs for downstream processing. Due to the blocking-after-service policy [26], when the intermediate buffer is full, a completed semi-finished item may remain at the upstream machine; see Equation (4).

2.3.4. Flow Dynamics of the Distribution Center

Downstream, the distribution center (DC) contains identical and perfectly reliable pooled parallel processing channels (M) that perform the final processing or dispatch operation before transferring completed items to three dedicated finished-goods buffers [27]. The pooled parallel processing channels (M) follow strict priority rules: type A items are processed first, followed by type B and then type C. No blocking is possible at the distribution center (DC) because processing halts whenever the corresponding finished-goods buffer is full. The pooled processing capacity is allocated according to the strict priority order $A > B > C$. A product type i is considered eligible for downstream processing when at least one corresponding semi-finished item is available in the intermediate buffer, $n_i(t) > 0$, and its finished-goods buffer has available capacity, $F_i(t) < F_i^{max}$. At each CTMC state, product type A is selected whenever it is eligible. Product type B is selected only when it is eligible and type A is not eligible, whereas product type C is selected only when it is eligible and neither type A nor type B is eligible. If none of the three product types is eligible, the distribution center remains inactive.

All M pooled processing channels are allocated collectively and exclusively to the selected product type. The channels are not divided among different product types, and lower-priority products are not processed simultaneously while a higher-priority product type is eligible. When product type i is selected, a downstream completion corresponds to the CTMC transition:

$$S_t(Q, n_i, F_i) \rightarrow S_{t+dt}(Q, n_i - 1, F_i + 1) \tag{10}$$

with rate $M \times \mu_{R_i}$.

At each completion event, exactly one item is transferred from the intermediate buffer to the corresponding finished-goods buffer. The allocation decision is re-evaluated immediately after every CTMC transition. Consequently, if a higher-priority product type becomes eligible, the complete pooled capacity is reassigned to it. Because processing times are exponentially distributed, the memoryless property makes it unnecessary to include residual processing times in the state vector.

2.3.5. Finished-Goods Buffer and Demand Dynamics

The customer-demand processes associated with products A, B, and C are modeled as mutually independent Poisson processes. For each product type $i \in \{A, B, C\}$, customer-demand arrivals occur at the product-specific rate λ_i, and each arrival concerns one unit of the corresponding finished product. The mutual-independence assumption means that the occurrence of a demand arrival for one product type does not affect the arrival processes or demand rates of the other product types.

If $F_i(t) > 0$, a demand arrival corresponds to the transition:

$$S_t(Q, n_i, F_i) \rightarrow S_{t+dt}(Q, n_i, F_i - 1) \tag{11}$$

with rate λ_i.

If $F_i(t) = 0$, the demand is lost.

2.4. Cardinality of State Space

Given the state definition introduced in Section 2.3.1, the total number of admissible system states (NS) can be derived explicitly. The upstream machine (Q) can be in six possible states: three productive states $Q \in \{A, B, C\}$ and three product-specific failure states $Q \in \{D_A, D_B, D_C\}$. The intermediate buffer (B_1) configuration is described by the vector (n_A, n_B, n_C), which is subject to the capacity constraint, $n_A + n_B + n_C \leq B_1$. The number of physical storage configurations is therefore given by $\binom{C+3}{3}$. The finished-goods buffers are characterized by their inventory levels (F_A, F_B, F_C), where each component takes an integer value between 0 and its corresponding maximum capacity. Hence, the number of feasible finished-goods buffer states equals $\prod_{i \in \{A,B,C\}} \left(F_i^{max} + 1\right)$.

Combining these components, the total number of states (NS) of the CTMC is:

$$NS = 6 \cdot \binom{C+3}{3} \cdot \prod_{i \in \{A,B,C\}} \left(F_i^{max} + 1\right) \tag{12}$$

which grows combinatorially with the buffer capacities [12]. This rapid growth motivates the use of numerical solution methods for the steady-state analysis.

2.5. Infinitesimal Generator Matrix and Steady-State Solution

Based on the state-space formulation and transition mechanisms described in the previous subsections, the stochastic behavior of the proposed multi-product push–pull production system is completely characterized by its infinitesimal generator matrix. Let $\Omega = \{s_1, s_2, \ldots, s_{NS}\}$ denote the set of all admissible system states, where $NS = \mid \Omega \mid$ is the state-space cardinality determined in Section 2.4. The infinitesimal generator matrix is defined as:

$$Z = \left[q_{ij}\right],\ i, j = 1, \ldots, NS \tag{13}$$

where each off-diagonal element $(q_{ij}, i \neq j)$ represents the transition rate from state (s_i) to state (s_j). These transition rates are determined by the feasible events defined in Section 2.3, including upstream production completions, setup changes, machine failures and repairs, downstream processing completions, and external demand arrivals. Accordingly,

$$q_{ij} = \begin{cases} r\left(s_i,\ s_j\right), \text{if a feasible transition from } s_i \text{ to } s_j \text{ exists,} \\ 0,\ \text{otherwise,} \end{cases}, i \neq j \tag{14}$$

where $r\left(s_i, s_j\right)$ denotes the corresponding event rate. The diagonal elements of the generator matrix are subsequently determined by:

$$q_{ii} = -\sum_{\substack{j=1 \\ j \neq i}}^{NS} q_{ij}, i = 1, \ldots, NS. \tag{15}$$

ensuring that each row of (Z) sums to zero.

To construct table Z, all admissible states are first generated according to the state definition and capacity constraints introduced in Section 2.3. Each state is assigned a unique index, after which the feasible events are examined sequentially. For every state (s_i), the corresponding transition conditions are evaluated, and whenever an event generates an admissible successor state (s_j), its transition rate is assigned to the respective off-diagonal element (q_{ij}). Once all feasible outgoing transitions have been identified, the corresponding diagonal element is calculated using Equation (15). This procedure provides a systematic algorithm for constructing the generator matrix for different feasible system configurations without requiring the transition structure to be specified manually.

The existence and uniqueness of the stationary distribution can be established from the structure of the proposed CTMC. First, B_1 and F_i^{max} are finite; therefore, the state space defined in Section 2.4 is finite. In addition, all transition rates are finite, implying a finite total exit rate from every state and ensuring that the CTMC is non-explosive.

Sufficient conditions for the existence of a unique stationary distribution are $M \geq 1$, strictly positive and finite rates μ_i, μ_{R_i}, λ_i, p_i, and r_i for every $i \in \{A, B, C\}$, and a strongly connected setup-transition graph. The latter condition means that every productive setting can be reached from every other productive setting through a sequence of setup transitions having positive rates.

Under these conditions, every enumerated state has a finite positive-rate path to the reference state $(A, 0, 0, 0, 0, 0, 0)$. If the upstream machine fails, a repair transition returns it to its corresponding productive state. Positive demand rates create available capacity in the finished-goods buffers, while successive downstream-processing and demand transitions provide a path through which all intermediate and finished-goods inventories can be reduced to zero. Once the upstream machine is not blocked, the strong connectivity of the setup-transition graph provides a path to production setting A.

It follows that every closed communicating class must contain the reference state. Therefore, the finite CTMC contains exactly one closed communicating class. This class is irreducible and positive recurrent. Any enumerated states outside this class are transient and have zero stationary probability. Consequently, the complete CTMC admits a unique stationary probability distribution:

$$\pi = [\pi_1, \pi_2, \ldots, \pi_{NS}] \tag{16}$$

where (π_i) denotes the long-run probability that the system occupies state (s_i). The steady-state probabilities are obtained by solving the global balance equations:

$$\pi \times Z = 0 \tag{17}$$

subject to the normalization condition:

$$\sum_{i=1}^{NS} \pi_i = 1, \pi_i \geq 0 \tag{18}$$

The resulting stationary distribution supplies the basis for evaluating the long-run behavior of the proposed production system. Once (π) has been determined, the product-specific and system-level performance measures can be computed as presented in the following section.

2.6. Performance Measures

The performance measures of the system illustrated in Figure 2 can be evaluated once the steady-state probability distribution of the CTMC described in this section is obtained. Let

$$p(Q, n_A, n_B, n_C, F_A, F_B, F_C) \tag{19}$$

denote the steady-state probability of the system being in state $(Q, n_A, n_B, n_C, F_A, F_B, F_C)$. The primary performance measures of this study are the mean throughput rate (THR_i), which is evaluated separately for each product type $i \in \{A, B, C\}$, and the total mean throughput rate (THR_{Total}) of the system, which is computed as the sum of the aforementioned mean throughput rates (THR_i). Additionally, the total work-in-process measure (WIP_{Total}) represents the long-run average number of semi-finished and finished items explicitly accounted for by the CTMC inventory variables, as defined in Section 2.6.2.

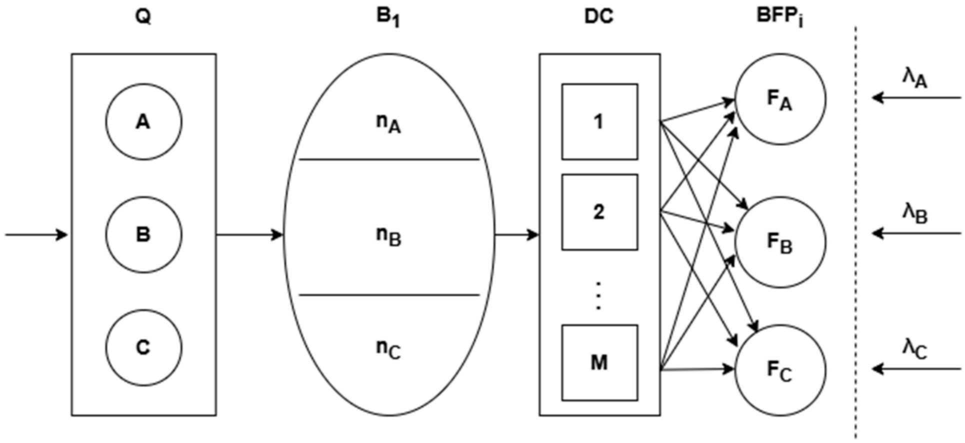

Figure 2. Push–pull system with multiple products, an intermediate buffer, and a distribution center with M pooled parallel processing channels. Arrows indicate the direction of material flow.

2.6.1. Mean Throughput Rates

For each finished product, $i \in \{A, B, C\}$, external customer-demand arrivals follow a Poisson process with rate λ_i. The three product-specific demand processes are assumed to be mutually independent. A demand is satisfied if and only if the corresponding finished-goods buffer is nonempty at the time of arrival; otherwise, the demand is lost. Consequently, the product's mean throughput rate equals the effective rate of satisfied demand. The mean throughput rates (THR_i) for the three product types $i \in \{A, B, C\}$ are therefore given by:

$$THR_A = \lambda_A \times \left[1 - \sum_{\substack{Q \in \{A,B,C,D_A,D_B,D_C\} \\ 0 \le n_A + n_B + n_C \le B_1 + 1 \\ 0 \le F_B \le F_B^{max} \\ 0 \le F_C \le F_C^{max}}} p(Q, n_A, n_B, n_C, 0, F_B, F_C) \right] \tag{20}$$

$$THR_B = \lambda_B \times \left[1 - \sum_{\substack{Q \in \{A,B,C,D_A,D_B,D_C\} \\ 0 \le n_A + n_B + n_C \le B_1 + 1 \\ 0 \le F_A \le F_A^{max} \\ 0 \le F_C \le F_C^{max}}} p(Q, n_A, n_B, n_C, F_A, 0, F_C) \right] \tag{21}$$

$$THR_C = \lambda_C \times \left[1 - \sum_{\substack{Q \in \{A,B,C,D_A,D_B,D_C\} \\ 0 \le n_A + n_B + n_C \le B_1 + 1 \\ 0 \le F_A \le F_A^{max} \\ 0 \le F_B \le F_B^{max}}} p(Q, n_A, n_B, n_C, F_A, F_B, 0) \right] \tag{22}$$

Each summation term represents the steady-state probability that the finished-goods buffer of the corresponding product type is empty, implying that incoming demand cannot be satisfied.

The total mean throughput rate (THR_{Total}) of the system is obtained as:

$$THR_{total} = THR_A + THR_B + THR_C \tag{23}$$

2.6.2. Average Inventory (Work-in-Process)

For the purposes of this study, the total average work-in-process measure ($\mathrm{WIP_{Total}}$) is operationally defined as the long-run average total number of semi-finished and finished items represented by the CTMC inventory variables. Specifically, $\mathrm{n_i}$ counts the semi-finished items of product type i that have not yet completed downstream processing, while $\mathrm{F_i}$ represents the corresponding finished-goods inventory. Under the aggregate pooled-capacity representation, an item selected for downstream processing remains included in $\mathrm{n_i}$ until its completion event occurs. At completion, $\mathrm{n_i}$ decreases by one and $\mathrm{F_i}$ simultaneously increases by one, as specified in Equation (10). Therefore, the distribution center does not constitute a separate inventory compartment, and no additional DC work-in-process term is required.

$$\mathrm{s} = (\mathrm{Q}, \mathrm{n_A}, \mathrm{n_B}, \mathrm{n_C}, \mathrm{F_A}, \mathrm{F_B}, \mathrm{F_C}) \tag{24}$$

denote a generic admissible state of the CTMC, and let (NS) be the corresponding state space defined in Section 2.4. The total average work-in-process ($\mathrm{WIP_{Total}}$) can then be expressed as:

$$\mathrm{WIP_{Total}} = \sum_{\mathrm{s} \in \mathrm{NS}} \left(\sum_{\mathrm{i} \in \{\mathrm{A,B,C}\}} (\mathrm{n_i(s)} + \mathrm{F_i(s)}) \right) \times \mathrm{p(s)} \tag{25}$$

Equation (25) therefore represents the long-run average number of items explicitly accounted for by the CTMC state variables. Since an item remains included in n_i during downstream processing and is transferred to F_i only upon completion, the measure accounts for the complete modeled inventory without requiring a separate distribution-center state variable or causing double counting.

2.7. Overall Solution Procedure

This section presents the steps that should be followed for the performance evaluation of the system described in Section 2. These steps are summarized in Algorithm 1.

Algorithm 1. Overall Solution Procedure for the Performance Evaluation of the Proposed Multi-Product Push–Pull System.

Step 1. Set the system parameters $\mathrm{B_1}$, $\mathrm{F_i^{max}}$, M, μ_i, $\mathrm{p_i}$, $\mathrm{r_i}$, α_ij, λ_i, $\mu_{\mathrm{R_i}}$.
Step 2. Generate the complete set of feasible system states and determine the state-space cardinality (NS), as described in Section 2.4
Step 3. Assign a unique index to each feasible state.
Step 4. Construct the infinitesimal generator matrix (Z) of dimension NS×NS.
- **Step 4.1.** Determine the feasible transitions from state ($\mathrm{s_i}$) to state ($\mathrm{s_j}$) according to the transition mechanisms described in Section 2.3
- **Step 4.2.** Assign the corresponding transition rates to the off-diagonal elements ($\mathrm{q_{ij}}$) of (Z).
- **Step 4.3.** Compute the diagonal elements of (Z) such that each row sums to zero.

Step 5. Solve the steady-state equations, $\pi \times \mathrm{Z} = 0$ subject to $\sum_{\mathrm{i}=1}^{\mathrm{NS}} \pi_\mathrm{i} = 1$.
Step 6. Compute the performance measures defined in Section 2.6.
- **Step 6.1.** Calculate the product-specific throughput rates ($\mathrm{THR_i}$).
- **Step 6.2.** Calculate the total system throughput ($\mathrm{THR_{Total}}$).
- **Step 6.3.** Calculate the total average work-in-process inventory ($\mathrm{WIP_{Total}}$).

Illustrative Construction of the State Space and Generator Matrix

To demonstrate how the transition mechanisms described in Section 2.3 are converted into the infinitesimal generator matrix, this subsection presents a detailed construction example based on the baseline configuration used in the numerical experiments. Specifically, $B_1 = 2$, $F_i^{max} = 2$, and $M = 2$, for $i \in \{A, B, C\}$, are considered together with the continuous-time transition-rate parameters reported in Table 1.

Table 1. Baseline continuous-time transition-rate parameters used in the numerical experiments.

Transition Rate	Product A	Product B	Product C
Production (μ_i)	1.5	1.2	1
Failure (p_i)	0.01	0.01	0.01
Repair (r_i)	0.1	0.1	0.1
Setup changes (α_i)	0.1	0.1	0.1
Demand (λ_i)	1	0.8	0.7
Service Rate (μ_{R_i})	1	1	1

The upstream-machine state Q is numerically encoded according to the ordered set

$$\mathcal{Q} = (A, B, C, D_A, D_B, D_C),$$

where A, B, and C represent the three operational production settings, whereas D_A, D_B, and D_C represent the corresponding failed states.

For each value of the upstream-machine state Q, the inventory components of the state vector must satisfy

$$n_A + n_B + n_C \le B_1 + 1$$

and

$$0 \le F_i \le F_i^{max}, i \in \{A, B, C\}.$$

For $B_1 = 2$, the effective upstream storage limit is $C = B_1 + 1 = 3$. Therefore, the number of feasible combinations of (n_A, n_B, n_C) is:

$$\binom{C+3}{3} = \binom{6}{3} = 20.$$

Because each finished-goods inventory variable F_i can take the values 0, 1, and 2, the number of inventory configurations associated with each upstream-machine state is:

$$H = \binom{C+3}{3} \prod_{i \in \{A,B,C\}} (F_i^{max} + 1) = 20 \times 3^3 = 540.$$

The complete state-space size is consequently:

$$NS = 6H = 3240.$$

The state vectors are generated first according to the ordered upstream-machine state set (A, B, C, D_A, D_B, D_C). Within each upstream-machine group, the remaining components are enumerated through ascending nested loops in the order:

$$n_A, |n_B, |n_C, |F_A, |F_B, |F_C.$$

More specifically, n_A ranges from 0 to $B_1 + 1$, n_B ranges from 0 to $B_1 + 1 - n_A$, and n_C ranges from 0 to $B_1 + 1 - n_A - n_B$. Each finished-goods inventory variable F_i ranges from 0 to F_i^{max}. The variable F_C changes most rapidly, followed by F_B, F_A, n_C, n_B, and n_A.

Each feasible combination generated through this nested enumeration is stored as a complete state vector:

$$s_i = (Q, n_A, n_B, n_C, F_A, F_B, F_C)$$

and is assigned a unique sequential index i, where $i = 1, \ldots, NS$. This index identifies both the position of the state in the ordered state-space array and the corresponding row and column of the infinitesimal generator matrix.

For example, the first and final states of the first upstream-machine block are:

$$s_1 = (A, 0, 0, 0, 0, 0, 0)$$

and

$$s_{540} = (A, 3, 0, 0, 2, 2, 2).$$

The first state of the second block is:

$$s_{541} = (B, 0, 0, 0, 0, 0, 0),$$

whereas the final state of the complete state space is:

$$s_{3240} = (D_C, 3, 0, 0, 2, 2, 2).$$

This enumeration establishes a unique correspondence between every admissible state vector and one row and column of the infinitesimal generator matrix.

Because the states are grouped according to Q, the generator has the following 6×6 block structure:

$$Z = \begin{bmatrix} Z_{A,A} & Z_{A,B} & Z_{A,C} & Z_{A,D_A} & 0 & 0 \\ Z_{B,A} & Z_{B,B} & Z_{B,C} & 0 & Z_{B,D_B} & 0 \\ Z_{C,A} & Z_{C,B} & Z_{C,C} & 0 & 0 & Z_{C,D_C} \\ Z_{D_A,A} & 0 & 0 & Z_{D_A,D_A} & 0 & 0 \\ 0 & Z_{D_B,B} & 0 & 0 & Z_{D_B,D_B} & 0 \\ 0 & 0 & Z_{D_C,C} & 0 & 0 & Z_{D_C,D_C} \end{bmatrix},$$

where each block has dimension $H \times H = 540 \times 540$.

The diagonal blocks contain the transitions that leave the upstream-machine state unchanged. These include production completions, pooled downstream-processing completions, customer-demand events, and the diagonal row-sum elements. The off-diagonal blocks $Z_{i,j}$, where $i, j \in \{A, B, C\}$ and $i \neq j$, contain the setup transitions. The blocks Z_{i,D_i} contain failure transitions, whereas the blocks $Z_{D_i,i}$ contain repair transitions. All other off-diagonal blocks are structurally zero because no event can directly generate the corresponding change in the upstream-machine state.

The generator is initialized as an $NS \times NS$ sparse zero matrix and is constructed row by row. Starting from each indexed state s_i, all feasible events defined in Section 2.3 are examined. If a feasible event generates the successor state s_j, the corresponding transition rate $r(s_i, s_j)$, as defined in Equation (14), is assigned to the off-diagonal generator element:

$$q_{ij} = r(s_i, s_j), i \neq j.$$

If more than one feasible event leads from s_i to the same successor state s_j, the corresponding event rates are added. After all feasible outgoing transitions from s_i have been inserted, the diagonal element is calculated as:

$$q_{ii} = -\sum_{\substack{j=1 \\ j \neq i}}^{NS} q_{ij},$$

ensuring that every row of Z sums to zero.

As a concrete example, consider the state s_i with index $i = 355$. According to the state-ordering procedure described above, this state is:

$$s_{355} = (A, 1, 1, 0, 0, 1, 0).$$

The upstream machine is operational and configured for product A. The intermediate storage contains one type-A item and one type-B item, whereas the finished-goods buffers contain zero type-A items, one type-B item, and zero type-C items. The total upstream inventory is:

$$n_A + n_B + n_C = 1 + 1 + 0 = 2 < B_1 + 1 = 3.$$

Therefore, the upstream machine is not blocked, and the following outgoing transitions are feasible.

First, a production completion of product A generates the transition

$$s_{355} = (A, 1, 1, 0, 0, 1, 0) \to s_{490} = (A, 2, 1, 0, 0, 1, 0)$$

with rate

$$q_{355,490} = \mu_A = 1.5.$$

Second, a failure of the upstream machine generates

$$s_{355} = (A, 1, 1, 0, 0, 1, 0) \to s_{1975} = (D_A, 1, 1, 0, 0, 1, 0)$$

with rate

$$q_{355,1975} = p_A = 0.01.$$

Third, a setup change from product A to product B generates

$$s_{355} = (A, 1, 1, 0, 0, 1, 0) \to s_{895} = (B, 1, 1, 0, 0, 1, 0)$$

with rate

$$q_{355,895} = \alpha_{AB} = 0.1.$$

Similarly, a setup change from product A to product C generates

$$s_{355} = (A, 1, 1, 0, 0, 1, 0) \to s_{1435} = (C, 1, 1, 0, 0, 1, 0)$$

with rate

$$q_{355,1435} = \alpha_{AC} = 0.1.$$

Both type-A and type-B items are available for downstream processing. However, because the distribution center follows the strict priority order A > B > C, the complete pooled downstream-processing capacity is allocated to product A. The resulting transition is

$$s_{355} = (A, 1, 1, 0, 0, 1, 0) \to s_{121} = (A, 0, 1, 0, 1, 1, 0)$$

with aggregate rate

$$q_{355,121} = M\mu_{R_A} = 2 \times 1 = 2.$$

Finally, because $F_B = 1$, customer demand for product B generates:

$$s_{355} = (A, 1, 1, 0, 0, 1, 0) \rightarrow s_{352} = (A, 1, 1, 0, 0, 0, 0)$$

with rate

$$q_{355,352} = \lambda_B = 0.8.$$

No customer-demand transitions are possible for products A and C because $F_A = F_C = 0$. Furthermore, product B is not selected for downstream processing because an eligible higher-priority product-A item is available.

Consequently, the nonzero off-diagonal elements of row 355 are:

$$\begin{gathered} q_{355,121} = 2, q_{355,352} = 0.8, q_{355,490} = 1.5, \\ q_{355,895} = 0.1, q_{355,1435} = 0.1, q_{355,1975} = 0.01. \end{gathered}$$

The corresponding diagonal element is:

$$\begin{aligned} q_{355,355} &= -\left(q_{355,121} + q_{355,352} + q_{355,490} + q_{355,895} + q_{355,1435} + q_{355,1975}\right) \\ &= -(2 + 0.8 + 1.5 + 0.1 + 0.1 + 0.01) \\ &= -4.51. \end{aligned}$$

All remaining elements of row 355 are zero. This example demonstrates how the feasible events associated with an individual state are translated directly into specific off-diagonal and diagonal elements of the infinitesimal generator matrix.

For the baseline configuration, the complete generator has dimension:

$$Z \in \mathbb{R}^{3240 \times 3240}$$

and contains 17,004 nonzero elements, including the diagonal elements. Its matrix density is:

$$\frac{17,004}{3240^2} \times 100\% = 0.162\%.$$

Each row contains at most eight feasible off-diagonal transitions in addition to its diagonal element, confirming the sparse structure of the matrix.

The stationary probability vector is obtained by solving the global balance equations and normalization condition presented in Equations (17) and (18). In the computational implementation, the generator is transposed, one balance-equation row is replaced by the normalization condition, and the corresponding right-hand-side element is set equal to one. The resulting linear system is then solved directly.

For the baseline configuration, the direct CTMC solution gives:

$$\begin{gathered} THR_A = 0.312449805, THR_B = 0.249959844, THR_C = 0.208299870, \\ THR_{Total} = 0.770709520 \text{ and } WIP_{Total} = 3.085751236. \end{gathered}$$

The infinity norm of the balance-equation residual is:

$$\| \pi Z \|_\infty = 1.27 \times 10^{-16},$$

whereas the normalization error is:

$$|\sum_{i=1}^{NS} \pi_i - 1| = 1.11 \times 10^{-16}.$$

These residuals are at the level of floating-point precision and confirm that the computed stationary distribution satisfies both the global balance equations and the normalization condition.

3. Results

In this section, numerical experiments evaluate the performance of the proposed multi-product push–pull system (Figure 2) and illustrate the impact of key system parameters on throughput and inventory levels under varying levels of disruptions and capacity constraints. The algorithm presented in Section 2 was implemented in MATLAB R2025b (The MathWorks, Inc., Natick, MA, USA). Configurations with intermediate-buffer capacity $B_1 \leq 20$ were evaluated on a desktop computer equipped with an Intel i5-6400 processor operating at 2.70 GHz, four processing cores, and 8 GB of RAM. For the computationally more demanding configurations with $B_1 > 20$, the independent experimental cases were distributed among higher-capacity workstations available in the university laboratory, and several configurations were executed concurrently. Each CTMC configuration was nevertheless generated and solved independently on a single workstation; no decomposition of an individual state space across different computers was employed. Because uniform timing logs were not retained for all concurrently executed configurations, the computational times reported in Appendix A are presented as retrospective order-of-magnitude estimates rather than as directly comparable benchmark measurements. The numerical analysis is based on the steady-state solution of the complete finite-state CTMC and the performance measures defined in Section 2.6.

3.1. Experimental Setup

A baseline parameter configuration is initially established to provide a common reference point for the numerical experiments. The selected parameter values define the production rates of the upstream machine, setup-transition rates, failure and repair characteristics, intermediate and finished-goods buffer capacities, downstream processing rates, external demand rates, and number of parallel downstream machines. Unless otherwise stated, these parameters remain fixed throughout the numerical analysis, while the parameter under investigation is varied within a predefined range. This experimental design allows the individual effect of each structural or operational factor on system performance to be identified while maintaining comparability among the examined scenarios. The baseline continuous-time transition-rate parameters used in the numerical experiments are summarized in Table 1. Unless otherwise stated, all rates are expressed in inverse time units.

For the baseline numerical experiments reported in Table 1, the setup-transition parameters are presented in compact form. Specifically, all off-diagonal transition-specific rates are set equal to 0.1, i.e., $\alpha_{AB} = \alpha_{AC} = \alpha_{BA} = \alpha_{BC} = \alpha_{CA} = \alpha_{CB} = 0.1$, while $\alpha_{ii} = 0$. This symmetric numerical parameterization constitutes a special case of the general sequence-dependent formulation.

Model Validation and Computational Performance

To independently validate the numerical solution of the proposed continuous-time Markov chain, we developed an event-driven discrete-event simulation (DES) model. Unlike the CTMC implementation, the simulation does not enumerate the state space or construct and solve the infinitesimal generator matrix. Instead, it advances the simulation clock directly from one stochastic event to the next and reproduces the production completions, setup transitions, machine failures and repairs, downstream processing, strict product-priority policy, and customer-demand events described in Section 2.

Consistent with the CTMC allocation rule, the M downstream processing channels are represented in the discrete-event simulation as a single pooled processing resource. At every simulation event, the complete pooled capacity is assigned exclusively to the highest-priority eligible product type according to the order $A > B > C$. The simulation does not permit the simultaneous processing of lower-priority products on otherwise idle channels while a higher-priority product type is eligible.

Accordingly, product type B is processed only when type A is not eligible, whereas product type C is processed only when neither type A nor type B is eligible. The allocation is re-evaluated after every event, and the selected product type is processed at the aggregate completion rate $M\mu_{R_i}$.

The validation experiment considered the baseline transition-rate parameters reported in Table 1. The additional structural parameters were set to $B_1 = 2$, $F_i^{Max} = 2$, $M = 2$, $p_i = 0.01$, and $r_i = 0.10$ for $i \in \{A, B, C\}$.

The simulation comprised 30 statistically independent replications. Each replication included a warm-up period of 5000 time units followed by an observation period of 100,000 time units. We used a reproducible base seed of 4,521,229.

The CTMC values for THR_A, THR_B, and THR_C were 0.3124, 0.2500, and 0.2083, respectively. The corresponding DES estimates were 0.312157, 0.250457, and 0.208660. The CTMC total throughput was 0.7707, compared with a DES estimate of 0.771274, while the CTMC and DES estimates of total work-in-process were 3.0858 and 3.088046, respectively. After rounding to four decimal places, the CTMC values reproduce the corresponding values reported in Table 2.

Table 2. Comparison of CTMC and discrete-event simulation results for the baseline configuration $B_1 = 2$, $F_i^{Max} = 2$, $M = 2$, $p_i = 0.01$, and $r_i = 0.10$ for $i \in \{A, B, C\}$.

Performance Measure	CTMC Values	DES Mean	95% Confidence Interval	Relative Deviation
THR_A	0.3124	0.312157	[0.310340, 0.313975]	0.094%
THR_B	0.2500	0.250457	[0.249078, 0.251836]	0.199%
THR_C	0.2083	0.208660	[0.207194, 0.210125]	0.173%
THR_{total}	0.7707	0.771274	[0.770085, 0.772463]	0.073%
$W.I.P_{Total}$	3.0858	3.088046	[3.082404, 3.093688]	0.074%

As shown in Table 2, all CTMC values were contained within the corresponding 95% simulation confidence intervals. The maximum relative deviation between the CTMC and DES results was only 0.199%, observed for THR_B. This close agreement provides independent support for the correctness of the state-space construction, transition mechanisms, strict-priority policy, pooled downstream-processing assumption, and steady-state performance calculations.

In this study, the term “exact” refers to the direct numerical solution of the complete finite-state CTMC. No state aggregation, decomposition, truncation, approximation, or simulation-based estimation is used to obtain the reported CTMC performance measures.

Computational tractability was additionally examined across the principal experimental configurations. Since the number of parallel downstream channels M and the positive transition-rate parameters affect the numerical values of the generator elements but not the admissible state space or its structural sparsity pattern, the computational diagnostics were organized according to the distinct intermediate-buffer capacities.

Across the complete range $0 \leq B_1 \leq 100$, the state-space cardinality increases from 648 to 29,500,848 states. The corresponding number of nonzero generator-matrix elements increases from 2916 to 188,764,416, whereas the matrix density decreases from 0.694444% to approximately 0.0000217%. The storage required for the MATLAB sparse representation of the generator matrix ranges from approximately 0.05 MiB to 3105.39 MiB, excluding the additional workspace and solver memory. These results demonstrate that the generator becomes increasingly sparse as buffer capacity grows, although the absolute computational burden increases substantially because of the combinatorial expansion of the state space.

For $B_1 \leq 20$, the indicative execution time ranges from less than one minute for the smallest configurations to approximately 30 min for the largest configurations in this group. For $21 \leq B_1 \leq 30$, the estimated execution time ranges from approximately 30 min to 3 h. The configurations with $B_1 = 50$ and $B_1 = 100$ require approximately 6–12 h and 1–3 days, respectively. These values are retrospective order-of-magnitude estimates derived from the state-space cardinalities and sparse-generator sizes and should not be interpreted as hardware-independent benchmark measurements.

The detailed computational characteristics are summarized in Appendix A. The results confirm that the sparse representation supports the analysis of the finite configurations examined in this study, while also showing that direct application of the complete-state approach becomes increasingly demanding as buffer capacity grows.

3.2. Impact of Intermediate Buffer Capacity

First, it is necessary to investigate the effect of the intermediate buffer capacity (B_1) on system performance. Table 3 presents the mean throughput rates (THR_i) per product, the total mean throughput rate (THR_{Total}) of the system, and total average work-in-process (WIP_{Total}) as functions of (B_1), with all other parameters held constant.

As expected, increasing buffer capacity (B_1) leads to higher throughput rates (THR_i) for all product types $i \in \{A, B, C\}$, and higher total mean throughput rate (THR_{Total}) of the system as well. However, the throughput gains diminish beyond a certain buffer size, indicating that the system transitions from a blocking-dominated regime to one in which downstream capacity and demand become the limiting factors.

At the same time, the total average work-in-process (WIP_{Total}) increases monotonically with (B_1), indicating greater inventory accumulation. These results highlight a fundamental trade-off between maximizing throughput and minimizing inventory, suggesting that moderate buffer capacities may be preferable from a managerial perspective [28].

Figure 3 shows that the two performance measures $(THR_{Total}, WIP_{Total})$ differ substantially, highlighting the rapid saturation of the system's total mean throughput rate (THR_{Total}) relative to the almost linear growth of the total average work-in-process (WIP_{Total}).

Figure 3 compares the effects of the intermediate-buffer capacity B_1 on the total mean throughput rate (THR_{Total}) and the total average work-in-process (WIP_{Total}). The total throughput increases rapidly for small buffer capacities and gradually approaches saturation as the downstream processing and demand rates become the dominant system constraints. In contrast, (WIP_{Total}) increases monotonically but does not exhibit a globally concave pattern. For low and moderate values of (B_1), additional buffer capacity reduces upstream blocking and improves coordination between the production stages. Once the throughput gains begin to diminish, further buffer expansion mainly allows additional intermediate inventory to accumulate without producing a proportional increase in productive output. Therefore, the marginal increase in WIP_{Total} is not required to decrease uniformly over the entire range of (B_1). Each point in Figure 3 corresponds to a separate discrete CTMC configuration, and the connecting lines are provided only as a visual guide.

Table 3. Impact of intermediate buffer capacity with transition rate parameters of Table 1 and structural parameters of the system $F_i^{Max} = 2, i \in \{A, B, C\}, M = 2$.

B_1	THR_A	THR_B	THR_C	THR_{total}	$W.I.P_{Total}$
0	0.2407	0.1926	0.1605	0.5937	1.5002
1	0.2866	0.2293	0.1911	0.7070	2.3503
2	0.3124	0.2500	0.2083	0.7707	3.0858
3	0.3321	0.2657	0.2214	0.8192	3.7742
4	0.3484	0.2787	0.2323	0.8594	4.4235
5	0.3623	0.2898	0.2415	0.8936	5.0347
6	0.3742	0.2994	0.2495	0.9231	5.6088
7	0.3846	0.3077	0.2564	0.9486	6.1468
8	0.3936	0.3149	0.2624	0.9708	6.6501
9	0.4014	0.3211	0.2676	0.9901	7.1197
10	0.4082	0.3266	0.2722	1.0070	7.5570
11	0.4142	0.3313	0.2762	1.0217	7.9628
12	0.4194	0.3356	0.2798	1.0348	8.3376
13	0.4240	0.3394	0.2830	1.0464	8.6821
14	0.4280	0.3428	0.2859	1.0567	8.9970
15	0.4316	0.3459	0.2885	1.0660	9.2829
16	0.4348	0.3486	0.2908	1.0742	9.5406
17	0.4376	0.3510	0.2929	1.0815	9.7707
18	0.4401	0.3531	0.2948	1.0880	9.9739
19	0.4423	0.3549	0.2965	1.0937	10.1508
20	0.4443	0.3565	0.2980	1.0988	10.3019
21	0.4461	0.3580	0.2994	1.1035	10.4296
22	0.4477	0.3593	0.3006	1.1076	10.5514
23	0.4491	0.3605	0.3017	1.1113	10.6675
24	0.4504	0.3616	0.3027	1.1147	10.7780
25	0.4516	0.3626	0.3036	1.1178	10.8832
26	0.4527	0.3635	0.3044	1.1206	10.9832
27	0.4536	0.3643	0.3052	1.1231	11.0783
28	0.4545	0.3650	0.3059	1.1254	11.1686
29	0.4553	0.3657	0.3065	1.1275	11.2544
30	0.4560	0.3663	0.3071	1.1294	11.3357
50	0.4635	0.3720	0.3120	1.1475	13.9518
100	0.4688	0.3762	0.3156	1.1606	19.8524

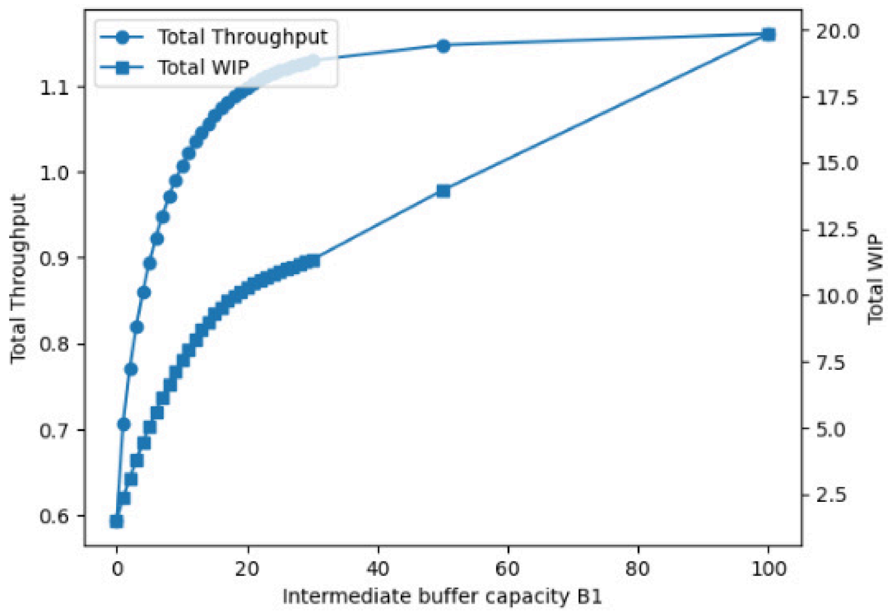

Figure 3. Effect of intermediate-buffer capacity (B_1) on the total mean throughput rate (THR_{Total}) and total average work-in-process (WIP_{Total}). The lines connect the results obtained for discrete CTMC configurations and are included as a visual guide.

3.3. Effect of Distribution Center Capacity

The effect of the number of parallel machines (M) at the distribution center is examined next. Table 4 reports the mean throughput rates (THR_i) per product, the total mean throughput rate (THR_{Total}) of the system, and total average work-in-process (WIP_{Total}) as functions of (M).

Table 4. Effect of distribution center capacity with transition rate parameters of Table 1 and structural parameters of the system, $F_i^{Max} = 2,\ i \in \{A, B, C\},\ B_1 = 2$.

M	THR_A	THR_B	THR_C	THR_{total}	$W.I.P_{Total}$
0	----	----	----	----	----
1	0.2552	0.2041	0.1701	0.6294	3.1068
2	0.3124	0.2500	0.2083	0.7707	3.0858
3	0.3300	0.2640	0.2200	0.8141	3.0679
4	0.3376	0.2701	0.2251	0.8328	3.0574
5	0.3416	0.2733	0.2277	0.8427	3.0512
6	0.3440	0.2752	0.2294	0.8487	3.0473
7	0.3457	0.2765	0.2304	0.8526	3.0447
8	0.3468	0.2774	0.2312	0.8554	3.0430
9	0.3476	0.2781	0.2317	0.8575	3.0418
10	0.3483	0.2786	0.2322	0.8590	3.0409
11	0.3488	0.2790	0.2325	0.8603	3.0403
12	0.3492	0.2793	0.2328	0.8613	3.0398
13	0.3495	0.2796	0.2330	0.8621	3.0394
14	0.3498	0.2798	0.2332	0.8628	3.0391
15	0.3500	0.2800	0.2334	0.8634	3.0389
16	0.3502	0.2802	0.2335	0.8639	3.0387
17	0.3504	0.2803	0.2336	0.8643	3.0386
18	0.3506	0.2805	0.2337	0.8647	3.0385
19	0.3507	0.2806	0.2338	0.8651	3.0384
20	0.3508	0.2807	0.2339	0.8654	3.0384
50	0.3521	0.2817	0.2347	0.8686	3.0384
100	0.3525	0.2820	0.2350	0.8695	3.0383

For small values of (M), the distribution center (DC) constitutes the primary bottleneck of the system, resulting in elevated inventory levels in the intermediate buffer (B_1) and frequent blocking of the upstream machine (Q). In particular, for high-priority products (k), each increase in the number of parallel machines (M) significantly increases both the mean throughput (THR_i) per product and the mean throughput rate (THR_{Total}) of the system, while the total average work-in-process (WIP_{Total}) decreases.

Beyond a certain point, however, additional distribution center (DC) capacity yields marginal benefits, as system performance becomes constrained by upstream production rates (μ_i) and external demand (λ_i). This observation suggests that excessive investment in downstream capacity may not be cost-effective unless accompanied by corresponding increases in upstream capability.

Figure 4 illustrates that the system's total mean throughput rate (THR_{Total}) increases quickly and approaches saturation for large values of parallel machines (M), whereas the total average work-in-process (WIP_{Total}) decreases slightly and stabilizes.

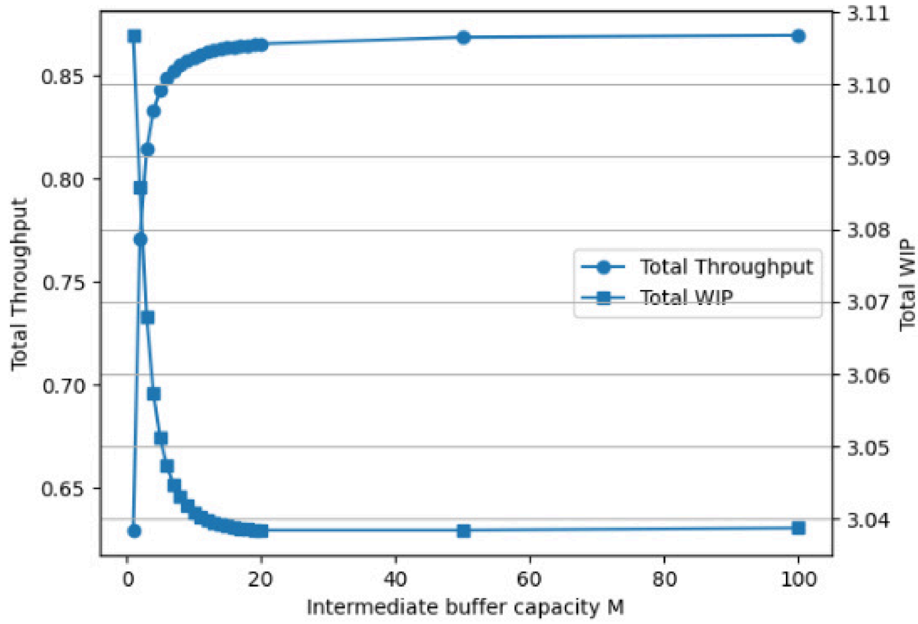

Figure 4. Effect of pooled parallel downstream processing channels (M) on total mean throughput rate (THR_{Total}) and work-in-process (WIP_{Total}).

3.4. Operational Implications of the Adopted Strict-Priority Rule

The numerical experiments presented in this study adopt a fixed strict-priority order $A > B > C$. This priority rule remains unchanged throughout all examined configurations, while Tables 3 and 4 investigate variations in the intermediate-buffer capacity B_1 and the number of pooled downstream processing channels M, respectively. Consequently, the product-specific throughput results reported in these tables represent system performance under the adopted strict-priority rule and do not constitute a comparative evaluation of alternative priority policies.

Under the adopted policy, the complete pooled downstream-processing capacity is assigned to the highest-priority eligible product type. Product type A is selected whenever it is eligible for downstream processing. Product type B is processed only when it is eligible and product type A is not eligible, whereas product type C is processed only when it is eligible and neither product type Anor product type B is eligible. Thus, a lower-priority product is never selected while an eligible higher-priority product is available.

Within the examined configurations, higher-priority products generally achieve higher mean throughput rates, whereas lower-priority products may receive fewer downstream-processing opportunities. In particular, product type C may experience limited access to the pooled downstream capacity under high system-load conditions because eligible type A and type B items are selected first. This service logic is consistent with the operation of priority-controlled multi-product systems discussed by Altiok et al. [29].

Nevertheless, the observed differences among THR_A, THR_B, and THR_C reflect the combined effects of the adopted priority order, product-specific production and demand rates, setup transitions, and inventory availability. They should therefore not be attributed exclusively to the priority rule. Since no alternative service discipline is employed as a comparative benchmark, the reported results should be interpreted as the operational implications of the adopted $A > B > C$ policy rather than as a formal comparison of different priority policies. A systematic comparison of alternative fixed, dynamic, or state-dependent priority policies constitutes a relevant direction for future research.

3.5. Effect of Machine Reliability

Finally, as the failure (p_i) and repair (r_i) rates of an upstream machine (Q) change, so does its reliability, which in turn affects the entire system. Thus, below, the system's sensitivity to different upstream machine (Q) performances is assessed. The results indicate that increased failure rates (p_i) significantly reduce the system's total mean throughput rate (THR_{Total}) and the total average work-in-process (WIP_{Total}), particularly when intermediate buffer (B_1) capacities are limited.

Notably, the negative impact of machine failures (p_i) is partially mitigated by larger intermediate buffers (B_1) and a higher number of parallel machines (M) in the distribution center (DC), thereby providing resilience against temporary production disruptions. This interaction underscores the importance of jointly considering reliability, buffering, and capacity decisions in the design of push–pull production systems.

The failure and repair rates jointly determine the operational availability of the upstream machine. The configurations presented in Table 5 simultaneously increase (p_i) and decrease (r_i), thereby representing progressively deteriorating joint reliability scenarios. These experiments allow the combined effect of upstream-machine reliability, intermediate-buffer capacity, and downstream-processing capacity to be examined. However, because (p_i) and (r_i) change simultaneously, the results of Table 5 cannot isolate the individual marginal effect of either parameter.

Table 5. Joint upstream-machine reliability scenarios obtained by simultaneously varying the failure and repair rates (p_i, r_i), with the remaining transition-rate parameters defined according to Table 1 and $F_i^{max} = 2, i \in \{A, B, C\}$.

Regime	Case	B_1	M	Failure Rate (p_i)	Repair Rate (r_i)	THR_{total}	$W.I.P_{Total}$
	1	2	2	0.002	0.30	0.8140	3.2784
High performance	2	2	2	0.003	0.27	0.8120	3.2687
	3	2	2	0.004	0.25	0.8099	3.2582
	4	2	2	0.007	0.18	0.7994	3.2093
Medium performance	5	2	2	0.008	0.16	0.7942	3.1861
	6	2	2	0.009	0.14	0.7875	3.1568
	7	2	2	0.014	0.09	0.7467	2.9808
Low performance	8	2	2	0.016	0.08	0.7280	2.9017
	9	2	2	0.018	0.07	0.7052	2.8059
	10	4	2	0.002	0.30	0.9119	4.7463
High performance	11	4	2	0.003	0.27	0.9095	4.7296
	12	4	2	0.004	0.25	0.9070	4.7114
	13	4	2	0.007	0.18	0.8944	4.6284
Medium performance	14	4	2	0.008	0.16	0.8881	4.5894
	15	4	2	0.009	0.14	0.8800	4.5403
	16	4	2	0.014	0.09	0.8303	4.2501
Low performance	17	4	2	0.016	0.08	0.8077	4.1215
	18	4	2	0.018	0.07	0.7800	3.9673
	19	2	4	0.002	0.30	0.8841	3.2681
High performance	20	2	4	0.003	0.27	0.8817	3.2573
	21	2	4	0.004	0.25	0.8791	3.2456
	22	2	4	0.007	0.18	0.8666	3.1917
Medium performance	23	2	4	0.008	0.16	0.8604	3.1662
	24	2	4	0.009	0.14	0.8525	3.1342
	25	2	4	0.014	0.09	0.8045	2.9438
Low performance	26	2	4	0.016	0.08	0.7828	2.8591
	27	2	4	0.018	0.07	0.7563	2.7571

The impact of the upstream machine (Q) reliability on system performance is examined through the system's total mean throughput rate (THR_{Total}) and the total average work-in-process (WIP_{Total}). Figure 5 shows the values of these metrics across different reliability regimes (High, Medium, and Low) and for two values of intermediate buffer (B_1).

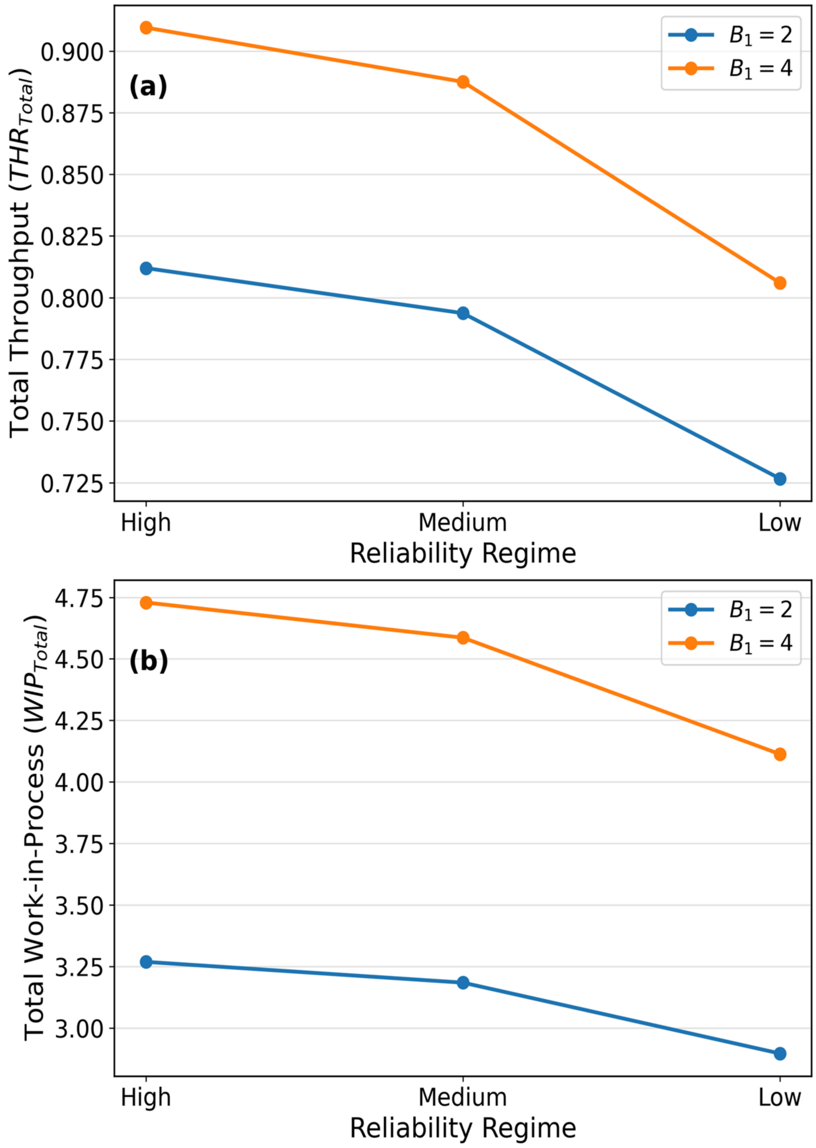

Figure 5. Effect of upstream machine reliability regimes on (**a**) total system throughput and (**b**) total work-in-process.

To distinguish the individual effects of the two reliability parameters, two additional one-factor-at-a-time experimental series were conducted. In both series, the structural parameters were maintained at $B_1 = 2$, $F_i^{max} = 2$, and $M = 2$, for $i \in \{A, B, C\}$, while all remaining transition-rate parameters were fixed at their baseline values from Table 1.

Table 6 isolates the effect of the upstream-machine failure rate. With $r_i = 0.10$ held constant, increasing (p_i) from 0.002 to 0.018 reduces (THR_{Total}) from 0.8072 to 0.7373. All three product-specific throughput rates decrease monotonically. The corresponding (WIP_{Total}) decreases from 3.2496 to 2.9361 because more frequent failures reduce the rate at which items enter the production and inventory system.

Table 7 isolates the effect of the upstream-machine repair rate. With $p_i = 0.01$ held constant, increasing (r_i) from 0.07 to 0.30 increases (THR_{Total}) from 0.7508 to 0.8029. The product-specific throughput rates also increase monotonically because faster repairs reduce the average duration of upstream-machine interruptions. The corresponding (WIP_{Total})

increases from 3.0049 to 3.2212 because improved upstream-machine availability increases the average production flow and the number of items present in the system.

Table 6. Controlled sensitivity analysis of the upstream-machine failure rate p_i, with $r_i = 0.10$, $B_1 = 2$, $F_i^{max} = 2$, and $M = 2$.

p_i	r_i	THR_A	THR_B	THR_C	THR_{total}	$W.I.P_{Total}$
0.002	0.10	0.3272	0.2618	0.2182	0.8072	3.2496
0.004	0.10	0.3234	0.2587	0.2156	0.7977	3.2072
0.010	0.10	0.3124	0.2500	0.2083	0.7707	3.0858
0.014	0.10	0.3055	0.2444	0.2037	0.7537	3.0093
0.018	0.10	0.2989	0.2391	0.1993	0.7373	2.9361

Table 7. Controlled sensitivity analysis of the upstream-machine repair rate r_i, with $p_i = 0.01$, $B_1 = 2$, $F_i^{max} = 2$, and $M = 2$.

p_i	r_i	THR_A	THR_B	THR_C	THR_{total}	$W.I.P_{Total}$
0.01	0.07	0.3044	0.2435	0.2029	0.7508	3.0049
0.01	0.10	0.3124	0.2500	0.2083	0.7707	3.0858
0.01	0.14	0.3180	0.2544	0.2120	0.7844	3.1422
0.01	0.25	0.3242	0.2594	0.2161	0.7997	3.2070
0.01	0.30	0.3255	0.2604	0.2170	0.8029	3.2212

4. Discussion

The numerical results demonstrate that the performance of the examined multi-product push–pull production system is determined by the interaction among storage capacity, downstream processing resources, product-priority policies, and upstream machine reliability. Rather than affecting system performance independently, these parameters influence the ability of the upstream and downstream stages to coordinate material flows and satisfy product-specific customer demand.

A particularly important finding concerns the role of intermediate buffer capacity. Increasing the shared buffer from $(C = 0)$ to $(C = 20)$ increases total throughput from 0.5937 to 1.0988, corresponding to an improvement of approximately 85.1%. However, further increasing the buffer to $(C = 100)$ results in a total mean throughput rate of only 1.1606, representing an additional improvement of approximately 5.6%. At the same time, total average work-in-process reaches 19.8524 units. These results reveal a clear throughput–inventory trade-off and indicate that the benefits of additional intermediate storage are characterized by diminishing returns. The buffer initially serves an important decoupling function between the two production stages by mitigating temporary imbalances between upstream production and downstream processing. Beyond a certain capacity, however, supplementary storage primarily increases the amount of material retained within the system rather than generating proportional improvements in productive output.

This observation has a direct implication for production-system design. Maximizing buffer capacity should not automatically be interpreted as an effective strategy for improving manufacturing performance. Instead, production managers should identify a capacity range in which sufficient decoupling between production stages is achieved without generating unnecessary WIP accumulation. In production applications, this distinction is important because larger buffers may involve additional space requirements, inventory-holding costs, material-handling activities, and capital tied up in semi-finished products. Although these economic factors are not explicitly incorporated into the present model, the simultaneous evaluation of throughput and WIP provides a quantitative basis for identifying configurations beyond which further capacity expansion becomes operationally less attractive.

A similar pattern is observed when downstream processing capacity is increased by adding parallel machines. Additional downstream resources initially improve the distribution center's ability to process semi-finished products and reduce the likelihood that downstream capacity constrains the overall production flow. Nevertheless, the marginal throughput improvement decreases as the number of parallel machines increases. Once sufficient downstream capacity has been installed, the limiting factor progressively shifts toward other components of the system, particularly upstream production capability, machine reliability, product availability in the intermediate buffer, and external demand. Consequently, simply adding parallel machines cannot generate unlimited improvements in system throughput.

This finding is particularly relevant to capacity investment decisions. The appropriate number of parallel downstream machines should be determined by considering the system's overall productive capacity rather than by maximizing the capacity of an individual production stage. Installing processing resources beyond the point at which another component becomes the dominant constraint may result in underutilized capacity without a corresponding increase in total throughput (THR_{Total}). The proposed CTMC framework can therefore support capacity-planning decisions by allowing alternative values of (M) to be evaluated before additional production resources are introduced.

The analysis of product-priority policies provides a different managerial perspective. Unlike buffer expansion or additional parallel capacity, priority rules primarily influence how existing production capacity is allocated among competing products. Changes in the processing sequence modify product-specific throughput (THR_i) and can therefore favor selected products without requiring physical modifications to the production system. This flexibility is particularly valuable in multi-product environments in which products differ in customer demand, strategic importance, service requirements, or profitability. Thus, a priority policy can be used as an operational decision lever for directing limited processing capacity toward products considered more important by production management.

However, the results also indicate that priority decisions should be evaluated at both the product and system levels. Improving the throughput of a high-priority product may reduce the processing opportunities available to lower-priority products. Thus, a policy that appears advantageous for a single product may lead to undesirable performance imbalances across the overall product portfolio. The product-specific throughput (THR_i) measures generated by the proposed framework enable explicit quantification of these effects and allow managers to compare alternative priority structures according to their operational objectives. Therefore, the preferred policy may depend on whether the objective is to maximize aggregate throughput, protect the availability of a strategically important product, or achieve a more balanced distribution of production capacity among product types.

Upstream machine reliability is another critical determinant of system performance. Since all three products depend on a common upstream production resource, failures at this stage influence the availability of semi-finished products for the entire downstream system. Improvements in machine reliability therefore help maintain material availability and reduce interruptions in the production flow. The performance benefit associated with reliability improvements also depends on the capacities of the remaining system components. When downstream processing or customer demand is constrained, further improvements in upstream reliability may yield progressively smaller gains in overall throughput.

From a managerial perspective, this result suggests that maintenance and reliability investments should be coordinated with capacity and inventory decisions. Improving the reliability of the upstream machine is particularly valuable when machine downtime represents a major restriction on system output. Conversely, when sufficient upstream availability has already been achieved, and the principal limitation lies downstream, supplementary reliability improvements may provide relatively limited production benefits. The analytical study can therefore be used to evaluate whether performance improvements are more effectively achieved through reliability enhancement, additional intermediate storage, or increased downstream processing capacity.

Collectively, the numerical experiments indicate that no single design parameter universally dominates system performance. Buffer capacity, downstream parallelization, priority policies, and machine reliability represent complementary managerial levers that address different sources of production-system limitation. Buffer capacity primarily supports the management of material flows between stages; parallel machines increase downstream processing capability; priority policies determine how this capability is allocated among competing products; and machine reliability affects the continuity of upstream material supply. Their effectiveness depends on the existing configuration of the production system and on which component currently restricts overall performance.

These findings reinforce the value of estimating production-system modifications through an integrated analytical framework. The proposed CTMC approach allows managers to perform "what-if" analyses before implementing physical or operational changes to the production system. Alternative buffer capacities, numbers of parallel machines, priority structures, and reliability levels can be evaluated under a common modeling environment, while their consequences can be assessed through product-specific throughput (THR_i), total throughput (THR_{Total}), and total average work-in-process (WIP_{Total}). The approach should therefore be interpreted not only as a performance-evaluation model but also as a decision-support tool for identifying configurations that provide an appropriate balance between productive output, inventory accumulation, capacity investment, and product-level operational priorities.

5. Conclusions

This paper presented an exact analytical framework for the performance evaluation of a multi-product push–pull manufacturing system with stochastic production, machine failures (p_i) and repairs (r_i), finite shared buffering (B_1), parallel downstream processing machines (M), and product-specific external demand (λ_i). The system is modeled as a continuous-time Markov chain with a high-dimensional discrete state space, capturing the complex interactions among production sequencing, buffering constraints, prioritization rules, and demand fulfillment.

The main methodological contribution of the study is the development of a general algorithm for systematically constructing the infinitesimal generator matrix for arbitrary system configurations. This approach enables the exact computation of steady-state probabilities and key performance measures, such as product-specific throughput rates (THR_i), total throughput rate (THR_{Total}) of the system, and total work-in-process inventory (WIP_{Total}).

The results demonstrate that the proposed modeling framework provides a powerful and flexible tool for analyzing and designing complex multi-product push–pull systems. This tool can support the implementation of production control mechanisms in which system parameters are dynamically adjusted based on real-time system conditions. Furthermore, the model offers valuable managerial insights into buffer sizing, capacity allocation, priority policy selection, and reliability management in manufacturing and supply chain environments characterized by product variety and demand uncertainty.

Several directions for future research naturally emerge from this study. First, the analytical framework can be extended to systems with more than three product types, multiple upstream machines, or more general network topologies, such as serial–parallel or re-entrant production structures. While such extensions would further increase the state-space size, they would enhance the model's applicability to real-world manufacturing systems.

Second, alternative service and control policies could be investigated. In particular, replacing strict priority rules at the distribution center (DC) with dynamic, state-dependent, or probabilistic scheduling policies may improve fairness among product types or increase overall system efficiency. Similarly, different push–pull boundary definitions and hybrid control mechanisms could be incorporated into the model.

Third, future work may relax some of the exponential and Poisson assumptions by incorporating phase-type or general distributions for processing times, setup times, and demand. Approximation or decomposition techniques could be combined with the exact Markovian framework to retain tractability while improving realism.

Finally, integrating economic performance measures—such as holding costs, shortage penalties, and setup costs—would enable optimization-based studies to jointly determine buffer capacities, production rates, and control parameters. Such extensions would further strengthen the model's usefulness as a decision-support tool for the design and operation of multi-product push–pull production systems and contribute to the design of resilient supply chains.

Author Contributions: Conceptualization, A.K. and A.C.D.; methodology, A.K.; software, S.K.; validation, M.A.M. and N.K.; formal analysis, N.K.; investigation, S.K.; data curation, A.C.D.; writing—original draft preparation, A.K.; writing—review and editing, A.C.D.; supervision, M.A.M.; project administration, A.C.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Computational Diagnostics

The state-space cardinality and structural sparsity of the generator matrix depend on the intermediate and finished-goods buffer capacities. Changes in the number of downstream processing channels or in the positive transition-rate parameters modify the numerical values of the generator elements but do not alter the number of admissible states or the structural nonzero pattern. Consequently, configurations sharing the same buffer capacities are grouped together in Table A1.

The sparse-storage values reported in Table A1 refer exclusively to the storage required for the MATLAB sparse representation of the generator matrix and exclude additional workspace, transpose, probability-vector, and solver-memory requirements. The execution-time ranges are retrospective order-of-magnitude estimates because several independent configurations were executed concurrently on different laboratory workstations and uniform timing logs were not retained. The values are therefore intended to indicate the approximate computational scale of the experiments rather than to constitute standardized hardware benchmarks.

For the baseline configuration with $B_1 = 2$, the complete generator contains 3240 states and 17,004 nonzero elements. The infinity norm of the balance-equation residual is 1.27×10^{-16}, while the normalization error is 1.11×10^{-16}. These values are at the level of floating-point precision and verify the numerical accuracy of the reported baseline stationary distribution.

Table A1. State-space cardinality, generator-matrix sparsity, sparse-storage requirements, and indicative computational-time ranges.

Experimental Configuration	B_1	States NS	Nonzero Elements	Density	Sparse Generator Storage	Indicative Execution Time
Small and moderate buffer configurations	0–20	648–327,888	2916–2,026,416	0.694444–0.001885%	0.05–33.42 MiB	<1 min–approximately 30 min
Intermediate-large buffer configurations	21–30	372,600–969,408	2,307,036–6,072,816	0.001662–0.000646%	38.05–100.06 MiB	Approximately 30 min–3 h
Large-buffer configuration	50	4,018,248	25,469,916	0.000158%	419.30 MiB	Approximately 6–12 h
Largest-buffer configuration	100	29,500,848	188,764,416	0.0000217%	3105.39 MiB	Approximately 1–3 days
Downstream-capacity and reliability experiments	2	3240	17,004	0.161980%	0.28 MiB	Generally <1 min per case
Reliability experiments with larger buffer	4	9072	50,592	0.061472%	0.84 MiB	Approximately 1–5 min per case

References

1. Florescu, A.; Barabas, S.A. Modeling and Simulation of a Flexible Manufacturing System—A Basic Component of Industry 4.0. *Appl. Sci.* **2020**, *10*, 8300. [CrossRef]
2. ElMaraghy, H.A. Flexible and reconfigurable manufacturing systems paradigms. *Int. J. Flex. Manuf. Syst.* **2005**, *17*, 261–276. [CrossRef]
3. Bajestani, M.S.; Mun, D.; Kim, D.B. Human-in-the-loop and large language models in smart manufacturing: Current applications, challenges, and perspectives. *J. Manuf. Syst.* **2026**, *86*, 913–941. [CrossRef]
4. Horr, A.M.; Milicic, S.; Blacher, D. AI-Driven Innovation in Manufacturing Digitalization: Real-Time Predictive Models. *Appl. Sci.* **2025**, *15*, 13225. [CrossRef]
5. Lee, H.L. Aligning supply chain strategies with product uncertainties. *Calif. Manag. Rev.* **2002**, *44*, 105–119. [CrossRef]
6. Lee, H.L.; Billington, C. Material management in decentralized supply chains. *Oper. Res.* **1993**, *41*, 835–847. [CrossRef]
7. Buzacott, J.A.; Shanthikumar, J.G. *Stochastic Models of Manufacturing Systems*; Prentice Hall: Englewood Cliffs, NJ, USA, 1993.
8. Gershwin, S.B. *Manufacturing Systems Engineering*; Prentice Hall: Englewood Cliffs, NJ, USA, 1994.
9. Li, J.; Meerkov, S.M. *Production Systems Engineering*; Springer: New York, NY, USA, 2009.
10. Kourepis, A.; Diamantidis, A.; Doios, K.; Papalamprou, K. Performance analysis of two stage flow lines with non-identical and unreliable parallel machines that produce good quality and defective products. *Int. J. Oper. Res.* **2024**, *50*, 332–363. [CrossRef]
11. Gu, X. Modeling and analysis of two-stage manufacturing systems with parallel machines and phase-type cycle time. *Manuf. Lett.* **2025**, *44*, 214–222. [CrossRef]
12. Magnanini, M.C.; Tolio, T. A Markovian model of asynchronous multi-stage manufacturing lines fabricating discrete parts. *J. Manuf. Syst.* **2023**, *68*, 325–337. [CrossRef]
13. Nemec, J.E. Diffusion and Decomposition Approximations of Stochastic Models of Multiclass Processing Networks. Ph.D. Thesis, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, 1999.
14. Colledani, M.; Gandola, F.; Matta, A.; Tolio, T. Performance evaluation of linear and non-linear multi-product multi-stage lines with unreliable machines and finite homogeneous buffers. *IIE Trans.* **2008**, *40*, 612–626. [CrossRef]
15. Mastrangelo, M.; Tolio, T.A.M. A hybrid method combining analytical and simulation models for performance evaluation of reconfigurable manufacturing systems. *J. Manuf. Syst.* **2024**, *76*, 259–280. [CrossRef]
16. Syrowicz, D. Decomposition Analysis of a Deterministic, Multiple-Part-Type, Multiple-Failure-Mode Production Line. Ph.D. Thesis, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, 1999.
17. Li, J.; Huang, N. Modeling and analysis of a multiple product manufacturing system with split and merge. *Int. J. Prod. Res.* **2005**, *43*, 4049–4066. [CrossRef]
18. Diamantidis, A.C.; Papadopoulos, C.T. Markovian analysis of a discrete material manufacturing system with merge operations, operation-dependent and idleness failures. *Comput. Ind. Eng.* **2006**, *50*, 466–487. [CrossRef]
19. Feng, W.; Zheng, L.; Li, J. Scheduling policies in multi-product manufacturing systems with sequence-dependent setup times and finite buffers. *Int. J. Prod. Res.* **2012**, *50*, 7479–7492. [CrossRef]

20. Kang, N.; Zheng, L.; Li, J. Analysis of multi-product manufacturing systems with arbitrary processing times. *Int. J. Prod. Res.* **2015**, *53*, 983–1001. [CrossRef]
21. Park, K.; Li, J. Improving productivity of a multi-product machining line at a motorcycle manufacturing plant. *Int. J. Prod. Res.* **2019**, *57*, 470–487. [CrossRef]
22. Chen, J.; Jia, Z.; Huang, L.; Dai, Y. Transient performance evaluation of flexible production lines with two Bernoulli machines and dedicated buffers. In *Proceedings of the 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), Hong Kong, China, 20–21 August 2020*; IEEE: New York, NY, USA; pp. 836–841.
23. Kourepis, A.; Diamantidis, A.C.; Koukoumialos, S.I. Exact analysis of a push–pull system with multiple non-identical retailers, a distribution center, and multiple identical unreliable suppliers with supply disruptions. *Oper. Res. Int. J.* **2022**, *22*, 4801–4827. [CrossRef]
24. Diamantidis, A.C.; Koukoumialos, S.I.; Vidalis, M.I. Performance evaluation of a push–pull merge system with multiple suppliers, an intermediate buffer and a distribution center with parallel machines/channels. *Int. J. Prod. Res.* **2016**, *54*, 2628–2652. [CrossRef]
25. Liberopoulos, G. Performance Evaluation of a Production Line Operated under an Echelon Buffer Policy. *IISE Trans.* **2018**, *50*, 161–177. [CrossRef]
26. Tan, B.; Gershwin, S.B. Analysis of a General Markovian Two-Stage Continuous-Flow Production System with a Finite Buffer. *Int. J. Prod. Econ.* **2009**, *120*, 327–339. [CrossRef]
27. Papadopoulos, H.T.; Heavey, C. Queueing Theory in Manufacturing Systems Analysis and Design: A Classification of Models for Production and Transfer Lines. *Eur. J. Oper. Res.* **1996**, *92*, 1–27. [CrossRef]
28. Liberopoulos, G. On the Trade-off between Optimal Order-Base-Stock Levels and Demand Lead-Times. *Eur. J. Oper. Res.* **2008**, *190*, 136–155. [CrossRef]
29. Altiok, T.; Shiue, G.A. Pull-Type Manufacturing Systems with Multiple Product Types. *IIE Trans.* **2000**, *32*, 115–124. [CrossRef]

https://doi.org/10.3390/app16189006