

AvPI: An Algorithm for Quantifying Average Parallelism in S3PR Petri Nets with Deadlock Prevention Policies

Luis O. Ventre^{1,2}, Orlando Micolini¹, Mauricio Ludemann¹, Gabriel Valenzuela¹

¹ UNC - FCEfN, Córdoba, Argentina

² luis.ventre@unc.edu.ar

Abstract. This research addresses the critical challenge of quantifying parallelism in embedded systems modeled with S3PR Petri nets, where deadlock prevention policies often impose restrictive constraints that limit concurrency. We introduce a novel algorithm to compute the Average Parallelism Index (AvPI), a metric evaluating the average number of concurrently executable tasks across all reachable states. The methodology leverages structural analysis of transition invariants and reachability graphs while excluding resource/buffer places to isolate actionable parallelism. Validated through a case study comparing deadlock-prone and deadlock-free configurations, results demonstrate that overly restrictive policies (ej., siphon control) reduce AvPI by 82% (to 0.9) and states by 99%, while selective control preserves high concurrency (AvPI=4.5953, 1017 states) without deadlocks. The AvPI bridges theoretical modeling and practical design, enabling early-stage optimization of resource allocation and reliability in critical systems like autonomous vehicles and industrial IoT. Future work will extend to timed Petri nets to incorporate temporal dynamics.

Keywords: S3PR Petri nets, embedded systems, deadlock prevention, AvPI, and parallelism quantification.

AvPI: Un algoritmo para cuantificar el paralelismo promedio en redes de Petri S3PR con políticas de prevención de interbloqueos

Resumen. Esta investigación aborda el desafío crítico de cuantificar el paralelismo en sistemas embebidos modelados con redes de Petri S3PR, donde las políticas de prevención de deadlock imponen restricciones que limitan la concurrencia. Presentamos un algoritmo innovador para calcular el Índice de Paralelismo Promedio (AvPI), métrica que evalúa el número promedio de tareas ejecutables concurrentemente en todos los estados alcanzables. La metodología integra análisis estructural de invariantes de transición y grafos de alcanzabilidad, excluyendo plazas de recursos/buffers para aislar el paralelismo accionable. Validado mediante un caso de estudio que compara configuraciones propensas y libres de deadlocks, los resultados demuestran que políticas sobre-restrictivas

(ej. control de sifones) reducen el AvPI en un 82% (a 0,9) y los estados en un 99%, mientras que controles selectivos preservan alta concurrencia (AvPI=4,5953; 1017 estados) sin bloqueos. El AvPI conecta modelado teórico y diseño práctico, permitiendo optimizar asignación de recursos y confiabilidad en etapas tempranas para sistemas críticos como vehículos autónomos e IoT industrial. Futuros trabajos se extenderán a redes temporizadas.

Palabras clave: Redes de Petri S3PR, sistemas embebidos, prevención de bloqueos, AvPI y cuantificación del paralelismo.

1 Introducción

El término "paralelismo", según la Real Academia Española (RAE), denota la cualidad de elementos que mantienen equidistancia constante sin cruzarse, extendiéndose metafóricamente a correspondencias entre entidades disímiles (De la Lengua Española, 2001). En sistemas embebidos, este concepto adquiere especificidad técnica: se refiere a la ejecución concurrente de múltiples tareas para optimizar eficiencia y rendimiento, contrastando con enfoques secuenciales (David Culler, 2017). Esta capacidad resulta crucial en aplicaciones que exigen respuestas en tiempo real y gestión eficiente de recursos limitados, como procesamiento y energía (Tanenbaum, 2016), especialmente ante la creciente complejidad de dominios tales como el Internet de las Cosas (IoT), vehículos autónomos y sistemas de control industrial (Wolf, 2010).

Para modelar dicha concurrencia, las Redes de Petri —en particular, los Sistemas de Procesos Secuenciales Simples con Recursos (S3PR)— ofrecen un marco formal que describe interacciones sincronizadas y competencia por recursos (Murata, 1989). No obstante, el diseño de estas redes conlleva riesgos sustanciales: políticas de disparo de transiciones y restricciones estructurales pueden generar **deadlocks** (bloqueos irreversibles) o limitar drásticamente el paralelismo (Abdul-Hussin, 2015), comprometiendo la capacidad de ejecución simultánea. Por ende, resulta imperativo cuantificar el impacto de tales políticas en el índice de paralelismo comparativo promedio (AvPI), métrica esencial que refleja el número medio de tareas procesables concurrentemente (Ezpeleta, Tricas, Garcia-Valles, & Colom, 2002; Lautenbach, 1987; Tricas, Garcia-Valles, Colom, & Ezpeleta, 2005).

En este contexto, se presenta un algoritmo innovador para determinar el **AvPI** en sistemas embebidos modelados con redes de Petri, continuando investigaciones previas (Z. Li, Wu, & Zhou, 2011; Ventre & Micolini, 2022). Este desarrollo complementa metodologías existentes para evaluar responsabilidad y cantidad de hilos (Ventre & Micolini, 2022), conformando un conjunto integral de herramientas de diseño temprano. La propuesta no solo optimiza la asignación de recursos y previene configuraciones erróneas derivadas de aproximaciones manuales (Navet & Simonot-Lion, 2017), sino que también impulsa la confiabilidad en aplicaciones críticas mediante la automatización de estimaciones de paralelismo.

2 Tesis Principal

Este trabajo aborda una brecha crítica en el análisis cuantitativo de sistemas embebidos modelados mediante RdP S3PR (Sistemas de Procesos Secuenciales Simples con Recursos), centrándose en la cuantificación objetiva del grado de paralelismo promedio (Average Parallelism Index , AvPI) y la evaluación del impacto de **políticas de disparo y criterios de desbloqueo** en esta métrica. La cuestión central se formula así: ¿Cómo puede medirse cuantitativamente el paralelismo efectivo en sistemas embebidos bajo restricciones de control, y de qué modo las políticas de disparo (estáticas/dinámicas) o los mecanismos de desbloqueo (ej. control basado en lugares clave) afectan dicho paralelismo? La respuesta a este interrogante permitirá resolver compensaciones fundamentales entre prevención de **deadlocks** y eficiencia concurrente (Ezpeleta, Colom, & Martinez, 1995; Marwedel, 2021), un desafío no resuelto en la síntesis de supervisores. Para ello, se postulan tres objetivos articulados:

1. Desarrollar el AvPI como métrica que cuantifique el número promedio de acciones ejecutables en paralelo a lo largo del espacio de estados alcanzables. Su fundamento teórico integra análisis estructural de redes de Petri (Murata, 1989) con métodos probabilísticos adaptados a espacios de estado complejos.
2. Validar la aplicabilidad del AvPI para evaluar configuraciones bajo políticas de disparo heterogéneas y estrategias de desbloqueo, identificando sobre-restricciones que, pese a garantizar ausencia de deadlocks (H. Liu, Xing, Wu, Zhou, & Zou, 2014), limiten injustificadamente la concurrencia.
3. Implementar un algoritmo innovador que automatice el cálculo del AvPI, complementando herramientas existentes para la evaluación de responsabilidad y cantidad de hilos (Ventre & Micolini, 2022). Este aporte consolida un marco metodológico integral para la toma de decisiones en etapas tempranas de diseño, alineado con enfoques de co-diseño hardware/software (SMIRI, DIMERAD, & EL AYEB, 2024) y mitigando errores derivados de aproximaciones manuales (Nivet & Simonot-Lion, 2017).

La propuesta se enmarca en la línea de investigación del autor (Eng. Ventre & Phd. Micolini, 2021; Ventre & Micolini, 2022), ofreciendo no solo un avance teórico en el modelado de la concurrencia, sino también un instrumento práctico para optimizar asignación de recursos, prevenir bloqueos y maximizar la eficiencia en sistemas embebidos multi-hilo de alta criticidad (ej. vehículos autónomos, IoT industrial).

2.1 Motivación, Relevancia y Revisión de la Literatura

El diseño de sistemas embebidos exige modelado riguroso del comportamiento dinámico para garantizar eficiencia y confiabilidad operativa. Las redes de Petri S3PR (Sistemas de Procesos Secuenciales Simples con Recursos) constituyen un marco formal ampliamente adoptado para representar sistemas concurrentes con competencia por recursos limitados (Murata, 1989). No obstante, un desafío crítico radica en la aparición de **deadlocks** (bloqueos irreversibles), estados donde ninguna transición puede dispararse, comprometiendo la funcionalidad del sistema (H. Liu et al., 2014). Para mitigar este riesgo, se implementan políticas de disparo (estáticas/dinámicas) y

restricciones estructurales que, paradójicamente, pueden inducir sobre-restricciones, limitando drásticamente el paralelismo y reduciendo la eficiencia del sistema (Roszkowska, 2004). Esta tensión entre prevención de bloqueos y optimización concurrente motiva la necesidad de métricas cuantitativas que evalúen objetivamente el impacto de tales políticas en el paralelismo efectivo, evitando diseños sub-óptimos donde los recursos se subutilizan (Marwedel, 2021).

La literatura evidencia avances significativos en modelado concurrente: las redes de Petri no deterministas (NT-Petri) capturan semánticas de disparo alternativas mediante transiciones no deterministas y agrupamiento en work clusters (Rawson & Rawson, 2022), mientras que extensiones temporizadas cuantifican desempeño mediante tiempos de ejecución y utilización de recursos (L. Hu et al., 2020). Paralelamente, enfoques basados en propiedades miden atributos estructurales como complejidad y acoplamiento en especificaciones concurrentes (Morasca, 1999). Pese a estas contribuciones, persiste una brecha metodológica en la cuantificación temprana del paralelismo promedio (Average Parallelism Index, AvPI). Enfoques existentes priorizan propiedades como vivacidad, seguridad o throughput (Ezpeleta et al., 1995), o se limitan a análisis particulares mediante simulaciones (H. Li, Sun, M., Zheng, W., Zeng, B. & Zeng, Y., 2020), careciendo de herramientas formales para comparar sistemáticamente la concurrencia inherente a configuraciones alternativas en fases iniciales de diseño (SMIRI et al., 2024). Esta limitación obstaculiza decisiones informadas en sistemas embebidos críticos (ej., vehículos autónomos, IoT industrial), donde equilibrar restricciones de control y eficiencia es imperativo (Zhou et al., 2023).

El presente trabajo aborda esta brecha mediante un algoritmo para calcular explícitamente el AvPI en redes S3PR, fundamentado en técnicas de álgebra lineal y teoría de grafos aplicadas al espacio de estados alcanzables. A diferencia de métodos basados en trazas específicas o métricas estáticas (ej. paralelismo máximo), este enfoque evalúa sistemáticamente el potencial de concurrencia a lo largo de todos los estados posibles, independientemente de ejecuciones particulares. Su innovación radica en integrar:

- Evaluación comparativa de políticas de disparo: Identificación de sobre-restricciones mediante contraste entre configuraciones.
- Criterios de desbloqueo: Análisis de mecanismos como control basado en lugares clave.
- Aplicabilidad temprana: Implementación en etapas iniciales de diseño mediante procesadores de Petri especializados.

Esta metodología optimiza la concurrencia promedio sin comprometer la corrección funcional, proporcionando un instrumento para maximizar la utilización de recursos en sistemas de misión crítica. Su relevancia trasciende el avance teórico al consolidar un marco cuantitativo para el co-diseño hardware/software, reduciendo errores de configuración manual y mejorando la confiabilidad sistémica.

2.2 Alcance: Limitaciones y Línea de Investigación Futura

El algoritmo propuesto presenta una limitación inherente a su marco conceptual actual: está diseñado para redes de Petri no temporizadas (Popova-Zeugmann, 2013), lo que implica que no considera el impacto de la dinámica temporal (ej., retardos de disparo o duraciones asociadas a transiciones) en el cálculo del índice de paralelismo promedio (AvPI). Esta restricción afecta particularmente su aplicabilidad en sistemas embebidos donde la temporalidad es un factor crítico para la eficiencia operativa, como en sistemas de control cíclico o entornos con restricciones de latencia estricta (Zhou et al., 2023). No obstante, esta limitación no invalida su utilidad en la evaluación de la estructura lógica de concurrencia durante fases preliminares de diseño, donde el análisis precede a la definición de características temporales y permite optimizar la coordinación estática de recursos (Marwedel, 2021).

Para superar esta barrera, se propone una línea de investigación futura centrada en extender el algoritmo a redes de Petri temporizadas. La estrategia fundamental implica:

1. Identificar sistemáticamente los marcados que involucren transiciones con retardos asociados.
2. Implementar un mecanismo de ponderación basado en parámetros temporales clave, como:
 - Tiempos de disparo de transiciones.
 - Frecuencia de activación en trazas de ejecución.
 - Impacto de cuellos de botella temporales en la concurrencia efectiva.

Esta adaptación permitiría que el AvPI refleje no solo el paralelismo potencial intrínseco, sino también la degradación inducida por la temporalidad, ofreciendo así una métrica holística del comportamiento dinámico. El enfoque se alinearía con metodologías establecidas para redes temporizadas (L. Hu et al., 2020), donde la integración de álgebra lineal y modelos de Markov podría cuantificar compensaciones entre concurrencia y restricciones temporales.

Esta extensión ampliaría sustancialmente la aplicabilidad del algoritmo, habilitando su uso en dominios como sistemas de tiempo real críticos y optimizando el co-diseño hardware/software en entornos con requisitos temporales estrictos.

3 Algoritmo propuesto

La metodología propuesta en esta investigación se estructura en las siguientes fases interrelacionadas que permiten evaluar sistemáticamente el AvPI en sistemas modelados con redes de Petri S3PR. La primera fase parte del sistema modelado con una red S3PR que puede presentar riesgo de deadlock, siguiendo los criterios estructurales establecidos en (Ezpeleta et al., 1995).

3.1 Construcción del grafo de alcanzabilidad

Mediante software especializado Tina (Toulouse, 2025), Pipe (Computing, 2002/3) o similar se determinan los marcados estables, es decir, todos los estados posibles sin llegar a la repetición de ciclos infinitos.

Identificación de invariantes de transición (IT)

Se deben identificar los IT de la red, cada invariante representa una secuencia mínima que lleva la red desde un marcado inicial hasta retornar al mismo marcado, lo que significa que el proceso ha cumplido un ciclo de acuerdo a las especificaciones, avanzando por las distintas etapas/estados del proceso compartiendo recurso y realizando acciones.

Obtención de plazas asociadas al IT

Para cada IT se debe obtener el conjunto de plazas asociadas a ese IT.

$$PI_i = U_{\forall t \in Inv} \bullet t \cup U_{\forall t \in Inv} t \bullet$$

Dónde: PI_i representa el conjunto de plazas asociadas al i ésimo IT.

Filtrado de plazas

Determinar las plazas relacionadas a acciones de cada IT. Para esto es necesario eliminar del conjunto PI_i , las plazas que son restricciones, recursos e idle(buffer):

$$PA_i = PI_i - \{PI_{restricciones_i} \cup PI_{recursos_i} \cup PI_{idle_i}\}$$

Dónde: PA_i representa el conjunto de plazas de acciones asociadas al i ésimo IT.

Luego necesitamos definir el conjunto PA , de todas las plazas de acción de la red, el cual es el equivalente a la unión de todas las plazas de acción de cada IT PA_i .

Estados Alcanzables

Del árbol de alcanzabilidad de la RdP, se debe obtener MA , el cual es el conjunto de todos los marcados posibles del conjunto de plazas PA .

Determinación del AvPI

Dado el conjunto de marcados MA

- Se debe operar sumando la cantidad de tokens de cada plaza en todos los estados y dividirla por la cantidad de estados estables del sistema.
- En esta instancia se obtendrá un promedio de tokens por plaza de acción en todos los marcados del sistema.

- Con este valor promedio para cada plaza de acción, sumando los promedios de tokens se obtendrá un índice promedio que determina el grado de paralelismo potencial de acciones del sistema modelado.

Caso de Aplicación y resultados

A continuación se detalla un caso de aplicación en donde con la finalidad de validar la aplicación de la metodología se parte de un sistema modelado con una red con deadlock, ver Fig. 1. Es importante mencionar que la red ha sido levemente complejizada a partir de la red expuesta en (Zhao & Hou, 2013).

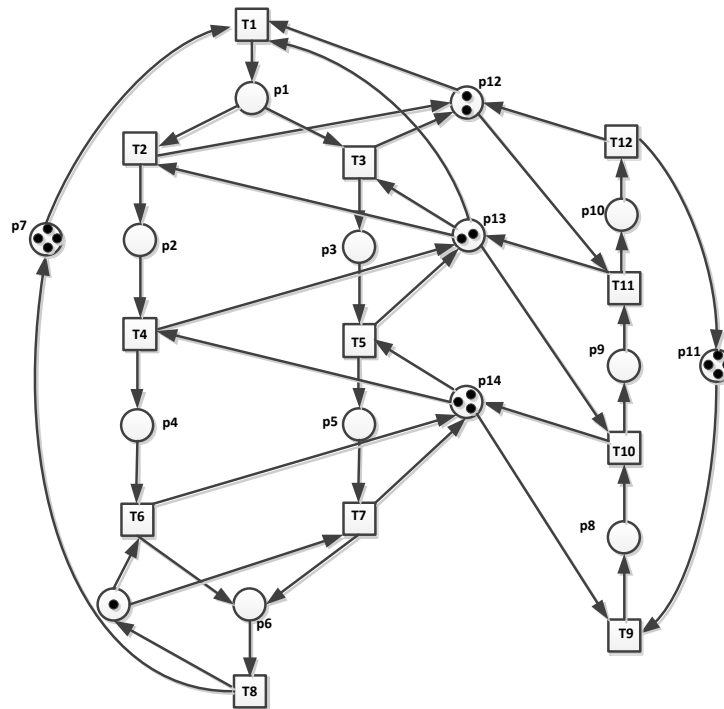


Fig. 1 Red de Petri original con deadlock

Se procede a desbloquear la misma con dos políticas de restricción distintas basadas respectivamente en el control de sifones mínimos y en políticas de priorización de transiciones, lo que permite contrastar diferentes enfoques de prevención de bloqueos ver

$IT_1 = \{T_9, T_{10}, T_{11}, T_{12}\}$ y Fig. 3. Estas propuestas de desbloqueo se limitan a solo el agregado de una plaza de restricción, a saber P_{16} , difiriendo en sus arcos de conexión en la red y su marcado inicial de tokens.

Luego se aplica la metodología propuesta a ambas redes y se analizan los resultados del AvPI para cada caso. Mediante herramientas de análisis formal como Petrina-

tor y PIPE, se construye el grafo de alcanzabilidad completo para cada variante del sistema. Este proceso considera únicamente los marcados estables, excluyendo aquellos estados transitorios que no contribuyen al comportamiento cíclico estable del sistema. La generación del espacio de estados se realiza hasta alcanzar saturación, evitando la repetición de ciclos infinitos mediante técnicas de poda automática.

Luego la metodología se centra en la identificación de IT's, que representan conjuntos mínimos de transiciones capaces de ejecutarse en ciclo completo (Murata, 1989). Este análisis estructural, realizado mediante métodos algebraicos sobre la matriz de incidencia, permite determinar los componentes fundamentales que contribuyen al paralelismo potencial del sistema, proporcionando una base matemática para la posterior cuantificación del índice.

Los IT de la red son:

$$IT_1 = \{T_9, T_{10}, T_{11}, T_{12}\}$$

$$IT_2 = \{T_1, T_2, T_4, T_6, T_8\}$$

$$IT_3 = \{T_1, T_3, T_5, T_7, T_8\}$$

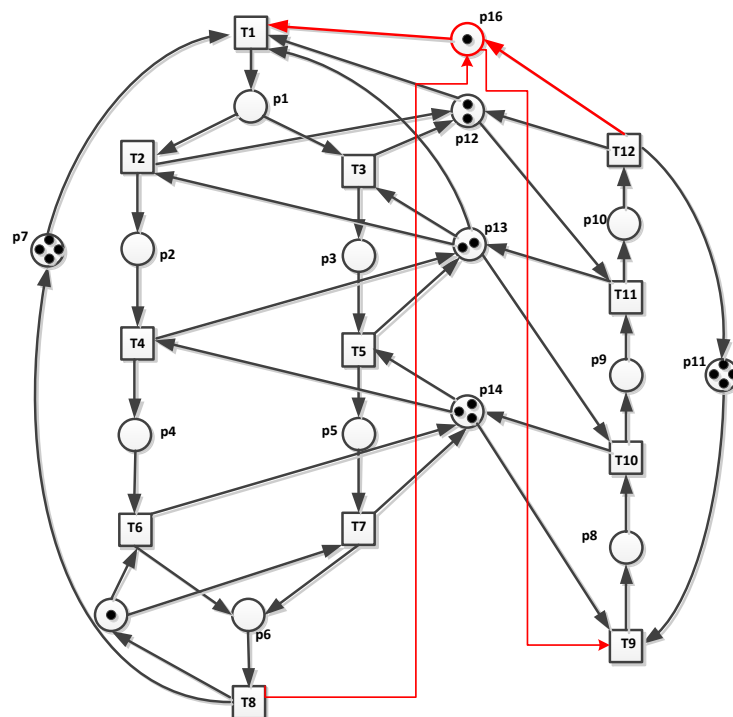


Fig. 2 Red de Petri solución propuesta 1

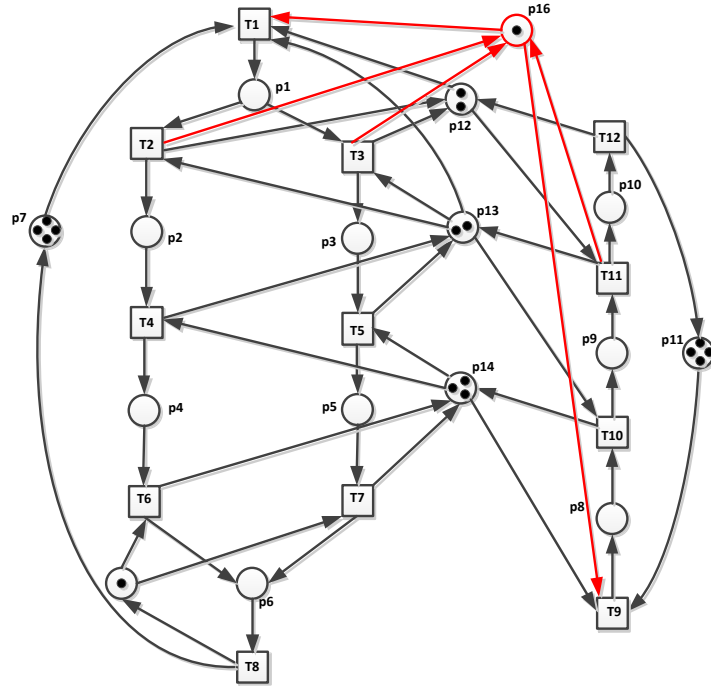


Fig. 3 Red de Petri solución propuesta 2

A continuación, se implementa el algoritmo que calcula el AvPI mediante un proceso en tres etapas. Inicialmente, se filtran las plazas correspondientes a recursos compartidos y buffers (idle places).

Plazas identificadas como Idle/buffer:

$$idle = \{P_7, P_{11}\}$$

Plazas identificadas como recursos /resources

$$resource\ restriction = \{P_{12}, P_{13}, P_{14}, P_{15}\}$$

Luego de identificar las plazas antes mencionadas y excluyéndolas del sistema se obtienen las plazas de acción de la red

$$PA = \{P_1, P_2, P_3, P_4, P_5, P_6, P_8, P_9, P_{10}\}$$

Posteriormente, se calcula el valor promedio de tokens en cada una de estas plazas de acción considerando todos los estados alcanzables, aplicando técnicas de muestreo

estadístico para redes complejas. Finalmente, la suma ponderada de estos promedios genera el AvPI, que cuantifica numéricamente la capacidad concurrente del sistema.

Al realizar el análisis con el grafo de marcados para la red original de la Fig. 1, se obtienen por un lado 1477 estados estables alcanzables y los siguientes promedios de tokens de todos esos estados para cada plaza:

Tabla 1. Promedio de tokens de estados alcanzables en red original

Plaza	Promedio de tokens
P_1	0.5223
P_2	0.3938
P_3	0.3938
P_4	0.5792
P_5	0.5792
P_6	0.4222
P_8	0.8627
P_9	0.5480
P_{10}	0.6441

Grado promedio de paralelismo

Operando obtenemos el AvPI de la red original (c/deadlock): 4.9452

Luego se realiza el análisis para la red de la solución 1 de la

$IT_1 = \{T_9, T_{10}, T_{11}, T_{12}\}$. Del análisis y la aplicación de la metodología es posible inferir que los invariantes de transición se mantienen así como también las plazas de acción y como es esperado solo se agrega una plaza de restricción P_{16} .

Al realizar el análisis con el grafo de marcados, se obtienen solo 10 estados estables alcanzables y los siguientes promedios de tokens de todos esos estados:

Tabla 2. Promedio de tokens de estados alcanzables en red solución 1

Plaza	Promedio de tokens
P_1	0.1000
P_2	0.1000
P_3	0.1000
P_4	0.1000
P_5	0.1000
P_6	0.1000
P_8	0.1000
P_9	0.1000
P_{10}	0.1000

3.2 Grado promedio de paralelismo

Operando obtenemos el AvPI de la red Solucion 1: 0.90

Luego se realiza el análisis para la red de la solución 2 de la Fig. 3. Del análisis y la aplicación de la metodología es posible inferir que los IT's se mantienen así como también las plazas de acción y restricciones.

Al realizar el análisis con el grafo de marcados se obtienen solo 1017 estados estables alcanzables y los siguientes promedios de tokens de todos esos estados:

Tabla 3. Promedio de tokens de estados alcanzables en red solución 2

Plaza	Promedio de tokens
P_1	0.2790
P_2	0.4273
P_3	0.4273
P_4	0.6729
P_5	0.6729
P_6	0.4204
P_8	0.5246
P_9	0.3890
P_{10}	0.7819

3.3 Grado promedio de paralelismo

Operando obtenemos el AvPI de la red Solucion 2: 4.5953

La fase final del estudio consiste en el análisis comparativo de los resultados obtenidos para las diferentes configuraciones del sistema. Este contraste multidimensional evalúa no solo la efectividad en la prevención de deadlocks, sino también el impacto en la concurrencia global, considerando métricas adicionales como el grado máximo de paralelismo y la densidad del grafo de alcanzabilidad. La comparación sistemática entre la red original (con deadlock) y las redes modificadas permite validar empíricamente las ventajas y limitaciones de cada política de control, aportando criterios objetivos para la toma de decisiones en el diseño de sistemas embebidos.

El marco analítico de esta investigación integra dos enfoques complementarios para evaluar rigurosamente las configuraciones del sistema. El análisis comparativo se fundamenta en el examen sistemático de métricas clave, particularmente el número de estados alcanzables y el AvPI. Esta aproximación permite cuantificar el trade-off entre las restricciones impuestas por las políticas de desbloqueo y la capacidad concurrente preservada, un aspecto crítico en el diseño de sistemas embebidos eficientes. La comparación se extiende más allá de valores absolutos, incorporando análisis de sensibilidad para identificar patrones de comportamiento en diferentes escenarios operativos.

La validación estructural constituye el segundo pilar metodológico, enfocándose en la preservación de las propiedades fundamentales del sistema mediante el examen de IT's. Esta verificación sigue los principios de conservación de comportamiento propuestos en (H. Hu, Zhou, Li, & Tang, 2012), garantizando que las modificaciones introducidas para prevenir deadlocks no comprometan la funcionalidad básica de la red. El proceso incluye tanto la verificación algebraica de invariantes lineales (Murata, 1989) como la evaluación de propiedades de vivacidad y boundedness, empleando para ello herramientas formales de análisis estructural implementadas en plataformas como PIPE (Computing, 2002/3) y Tina (Toulouse, 2025). Esta doble capa de validación asegura que las soluciones propuestas no solo mejoren la confiabilidad del sistema, sino que mantengan su corrección funcional frente a especificaciones de diseño originales.

4 Técnicas de Análisis

El marco metodológico integra dos enfoques complementarios para evaluar configuraciones de sistemas embebidos modelados con redes de Petri S3PR. El análisis comparativo examina métricas clave —número de estados alcanzables e índice de paralelismo promedio (AvPI)— siguiendo metodologías establecidas para sistemas discretos (Basile, 2023). Esta aproximación cuantifica el trade-off entre restricciones de control y capacidad concurrente, incorporando análisis de sensibilidad para patrones de comportamiento en escenarios operativos heterogéneos (SMIRI et al., 2024; Wang, 2023). Paralelamente, la validación estructural verifica la preservación de propiedades fundamentales mediante invariantes de transición, garantizando que las modificaciones para prevenir deadlocks no comprometan la funcionalidad básica (H. Liu et al., 2014). Este proceso combina verificación algebraica de invariantes lineales (Murata, 1989) con evaluación de vivacidad y boundedness (Y. Liu, Li, & Li, 2023), utilizando herramientas como PIPE y Tina para asegurar corrección funcional frente a especificaciones de diseño.

4.1 Resultados Experimentales

La aplicación metodológica revela características críticas del comportamiento en redes S3PR:

- Configuración original : Exhibe alto paralelismo (AvPI = 4.9452 en 1477 estados alcanzables), pero tiende a deadlocks , invalidando su aplicabilidad práctica (Ezpeleta et al., 1995).
- Enfoque restrictivo (control de sifones) : Elimina deadlocks pero reduce drásticamente los estados alcanzables (10) y el AvPI (0.9), evidenciando sobre-restricción que limita la concurrencia (Basile, 2023) (H. Liu et al., 2014) .
- Control selectivo : Mantiene un equilibrio óptimo, con AvPI = 4.5953 y 1017 estados alcanzables sin bloqueos, preservando eficiencia operativa (SMIRI et al., 2024).

Este contraste demuestra empíricamente el trade-off inherente al diseño de sistemas concurrentes: soluciones maximalmente restrictivas garantizan seguridad pero generan ineficiencias, mientras enfoques balanceados optimizan paralelismo sin sacrificar confiabilidad (Wang, 2023).

4.2 Discusión e Implicaciones

El AvPI emerge como indicador cuantitativo esencial para evaluar compensaciones entre seguridad y eficiencia, identificando diseños sobre-restringidos donde la prevención de deadlocks limita injustificadamente la concurrencia (Basile, 2023; Zhou et al., 2023). En sistemas embebidos críticos (ej., control industrial, vehículos autónomos), políticas restrictivas mal optimizadas pueden causar:

- Extensión de tiempos de respuesta.
- Subutilización de recursos computacionales.
- Riesgos operativos en entornos de tiempo real duro (SMIRI et al., 2024).

La integración del AvPI en etapas tempranas de diseño —alineada con principios shift-left (Wang, 2023)— permite ajustes estructurales viables y económicos, reduciendo errores de configuración manual y acelerando ciclos de desarrollo.

5 Conclusión

Este estudio desarrolla un marco metodológico innovador para cuantificar el paralelismo en sistemas embebidos mediante redes S3PR, introduciendo el AvPI como métrica para evaluar compensaciones entre restricciones de control y eficiencia concurrente. Sus contribuciones clave son:

1. Herramienta analítica: Cálculo del AvPI basado en análisis del grafo de alcanzabilidad e invariantes de transición, excluyendo plazas asociadas a recursos o buffers.
2. Optimización del diseño: Integración con algoritmos existentes para evaluación de hilos, consolidando un conjunto de herramientas para diseño multi-hilo.
3. Validación empírica: Demostración de que soluciones balanceadas (ej., control selectivo) preservan paralelismo sin comprometer confiabilidad.

Pese a limitaciones inherentes (naturaleza estática, explosión combinatoria, ausencia de temporalidad), el trabajo establece bases para extensiones futuras:

- Incorporación de redes temporizadas para modelar dinámicas operativas.
- Técnicas de abstracción de espacios de estado grandes (Y. Liu et al., 2023).
- Estrategias adaptativas de control en sistemas complejos.

Esta investigación puentea teoría de redes de Petri y necesidades prácticas en sistemas embebidos contemporáneos, ofreciendo un instrumento cuantitativo para toma de decisiones en dominios críticos donde confiabilidad y eficiencia coexisten en tensión permanente (SMIRI et al., 2024).

References

- Abdul-Hussin, M. (2015). *Design of a Petri net based deadlock prevention policy supervisor for S3PR*. Paper presented at the 2015 6th International Conference on Intelligent Systems, Modelling and Simulation.
- Basile, F., Chiacchio, P. & De Tommasi, G. (2023). Performance Evaluation in Discrete-Event Systems. *Springer Nature, Cham, Suiza.*, 498, 153-184.
- Computing, I. C. L. D. o. (2002/3). Platform Independent Petri Net Editor PIPE. Retrieved from <https://pipe2.sourceforge.net/>
- David Culler, J. P. S., Anoop Gupta Ph.D. (2017). *Parallel Computer Architecture: A Hardware/Software Approach*.
- De la Lengua Española, D. (2001). Real academia española.
- Eng. Ventre, L. O., & Phd. Micolini, O. (2021). *Extended Petri Net Processor and Threads Quantity Determination Algorithm for Embedded System*. In E. J. Pesado P. (Ed.), *Communications in Computer and Information Science CCIS* (Vol. 1409). doi:https://doi.org/10.1007/978-3-030-75836-3_14
- Ezpeleta, J., Colom, J. M., & Martinez, J. (1995). A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems. *IEEE Trans on Robotics and Automation*, 11, 173–184.
- Ezpeleta, J., Tricas, F., Garcia-Valles, F., & Colom, J. M. (2002). A banker's solution for deadlock avoidance in FMS with flexible routing and multiresource states. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 18(4), 621-625.
- Hu, H., Zhou, M., Li, Z., & Tang, Y. (2012). Deadlock-free control of automated manufacturing systems with flexible routes and assembly operations using Petri nets. *IEEE Transactions on Industrial Informatics*, 9(1), 109-121.
- Hu, L., Liu, Z., Hu, W., Wang, Y., Tan, J., & Wu, F. (2020). Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. *Journal of Manufacturing Systems*, 55, 1-14.
- Lautenbach, K. (1987). *Linear algebraic techniques for place/transition nets*. Paper presented at the Petri Nets: Central Models and Their Properties: Advances in Petri Nets 1986, Part I Proceedings of an Advanced Course Bad Honnef, 8.–19. September 1986.
- Li, H., Sun, M., Zheng, W., Zeng, B. & Zeng, Y. (2020). Simulation-Based Concurrency Analysis for Non-Intrusive Scheduling in Embedded Systems. *Proceedings of the International Conference on Embedded Software (EMSOFT)*.
- Li, Z., Wu, N., & Zhou, M. (2011). Deadlock control of automated manufacturing systems based on Petri nets—A literature review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4), 437-462.
- Liu, H., Xing, K., Wu, W., Zhou, M., & Zou, H. (2014). Deadlock prevention for flexible manufacturing systems via controllable siphon basis of Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3), 519-529.
- Liu, Y., Li, X., & Li, Z. (2023). Control strategy of discrete event systems modeled by labeled Petri nets based on transition priority. *IEEE Access*, 11, 45442-45455.
- Marwedel, P. (2021). *Embedded system design: embedded systems foundations of cyber-physical systems, and the internet of things*: Springer Nature.

- Morasca, S. (1999). *Measuring attributes of concurrent software specifications in petri nets*. Paper presented at the Proceedings Sixth International Software Metrics Symposium (Cat. No. PR00403).
- Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, Vol. 77, No. 74, pp. 541-580.
- Navet, N., & Simonot-Lion, F. (2017). *Automotive embedded systems handbook*: CRC press.
- Popova-Zeugmann, L. (2013). Time and Petri Nets. *Springer*.
- Rawson, M., & Rawson, M. G. (2022). *Petri Nets for Concurrent Programming*. Paper presented at the 2022 IEEE/ACM Fifth Annual Workshop on Emerging Parallel and Distributed Runtime Systems and Middleware (IPDRM).
- Roszkowska, E. (2004). Supervisory control for deadlock avoidance in compound processes. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 34(1), 52-64.
- SMIRI, K., DIMERAD, R. B., & EL AYEB, F. (2024). HARDWARE/SOFTWARE CO-DESIGN IN MULTIPROCESSORS EMBEDDED SYSTEMS AND IOT. *Journal of Theoretical and Applied Information Technology*, 102(2).
- Tanenbaum, A. S. (2016). *Structured computer organization*: Pearson Education India.
- Toulouse, L.-C. U. d. (2025). Time Petri Net Analyzer TINA. Retrieved from <https://projects.laas.fr/tina/index.php>
- Tricas, F., Garcia-Valles, F., Colom, J. M., & Ezpeleta, J. (2005). *A Petri net structure-based deadlock prevention solution for sequential resource allocation systems*. Paper presented at the Proceedings of the 2005 IEEE international conference on robotics and automation.
- Ventre, L. O., & Micolini, O. (2022). *Algoritmos para determinar cantidad y responsabilidad de hilos en sistemas embebidos modelados con Redes de Petri S3PR*. Paper presented at the XXVII Congreso Argentino de Ciencias de la Computación (CACIC)(Modalidad virtual, 4 AL 8 DE OCTUBRE DE 2021.).
- Wang, Y., Jiang, Y., Chang, W. & Zeng, Y. (2023). Sensitivity Analysis for Real-Time Concurrent Systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 22(5), 1-27.
- Wolf, W. (2010). *High-performance embedded computing: architectures, applications, and methodologies*: Morgan Kaufmann.
- Zhao, M., & Hou, Y. (2013). An iterative method for synthesizing non-blocking supervisors for a class of generalized Petri nets using mathematical programming. *Discrete Event Dynamic Systems*, 23, 3-26.
- Zhou, F., Cao, Z., Ge, Q., Zhang, Q., Zhao, D., Liu, S., & Chen, Y. (2023). A real-time phase processing system for phase sensitive optical time domain reflectometer. *Review of Scientific Instruments*, 94(1).