



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERIA DE SISTEMAS – INVESTIGACIÓN DE OPERACIONES

DISEÑO DE UNA HEURÍSTICA PARA RESOLVER UN PROBLEMA DE SECUENCIACIÓN
EN LA INDUSTRIA DE AUTOMÓVILES

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
ALEJANDRA ANGÉLICA RUVALCABA LÓPEZ

TUTOR:
DRA. IDALIA FLORES DE LA MOTA,
FACULTAD DE INGENIERIA

MÉXICO, D. F. OCTUBRE 2013.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. Aceves García Ricardo
Secretario: M.I. Wellens Purnal Ann Godelieve
Vocal: Dra. Flores De La Mota Idalia
1er. Suplente: M.I. Soler Anguiano Francisca Irene
2do. Suplente: Dra. Segura Pérez Esther

MÉXICO, CIUDAD UNIVERSITARIA

TUTOR DE TESIS:

DRA. IDALIA FLORES DE LA MOTA

FIRMA

Diseño de una heurística para resolver un problema de secuenciación en la industria de automóviles

Alejandra A. Ruvalcaba López

INDICE

1. INTRODUCCION	1
1.1 Objetivos.....	1
1.2 Alcances y Limitaciones.....	2
1.3 Organización del Trabajo.....	2
2. MARCO TEÓRICO.....	1
2.1. Logística	1
2.2. Sistemas de Producción.....	2
2.3. Modelos de Toma de Decisiones.....	2
2.4. Secuencia y Programación de Tareas	3
2.5. Estado del arte. Problema de secuenciación de la industria del automóvil	5
3. REDES DE PETRI.....	2
3.1 Descripción de una Red de Petri (RdP)	2
3.2 Definición Formal de las Redes de Petri (RdP).....	3
3.3 Dinámica y representación de las Redes de Petri (RdP).....	4
3.4 Redes de Petri Coloreadas (RPC).....	6
3.5 Descripción de una Red de Petri Coloreada (RPC).....	8
3.6 Definición Formal de las Redes de Petri Coloreadas (RPC)	9
3.7 Dinámica y representación de las Redes de Petri Coloreadas (RPC).....	10
3.8 Redes de Petri Coloreadas Temporales	12
3.9 Análisis	13

4. MODELADO DE SISTEMAS LOGÍSTICOS EN REDES DE PETRI.....	16
4.1 Descripción del Problema de Secuencia.....	16
4.2 Comportamiento de la Secuencia	17
4.3 Sistema de reparación	19
4.4 Modelo Gráfico.....	21
5. PROBLEMA DE SECUENCIA. ALGORITMO DE REPARACIÓN	28
5.1 Definición Formal	28
5.2 Algoritmo	29
5.3 Observaciones	35
6. CASO DE ESTUDIO Y RESULTADOS.....	36
6.1 Representación del Sistema. Método de Intercambio	36
6.2 Herramienta de Modelado	45
6.3 Resultados.....	49
6.4 Codificación del Algoritmo.....	52
7. CONCLUSIONES	60
BIBLIOGRAFIA	62

Capítulo 1

INTRODUCCION

La alta competencia que se presenta actualmente en el mercado, ha obligado a las empresas a desarrollar investigación en las diferentes áreas administrativas y de procesos de producción con el objetivo de presentarse ante el mercado como una empresa competitiva sin descuidar la rentabilidad de la misma.

En consecuencia, la búsqueda de herramientas y metodologías que apoyen de forma eficiente en el desarrollo de las actividades diarias, constituye parte fundamental del desarrollo y crecimiento de la empresa en el sector, en particular dentro de la industria manufacturera, un elemento muy importante es el análisis de los sistemas de producción, debido a la influencia que tiene una correcta planificación de las actividades de transformación sobre los parámetros de rendimiento del sistema.

Al hablar de los sistemas de producción, es necesario recurrir a la toma de decisiones, ésta abarca una gran diversidad de aspectos que consideran desde los insumos y actividades realizadas dentro del sistema como otros aspectos más estratégicos de dimensionamiento de recursos. Si se considera el caso particular de la producción multiproducto, uno de los problemas de decisión radica en la secuencia de elementos o piezas a ser transformados en las diferentes estaciones de trabajo.

Al buscar soluciones a través de métodos exactos a problemas de secuenciación cuya naturaleza es combinatoria, es importante señalar que estos problemas están definidos como NP-completo, para ellos, la búsqueda de soluciones óptimas requiere un tiempo computacional que depende del tamaño del problema debido al elevado número de combinaciones a evaluar.

Tomando en cuenta la dificultad que se presenta con los métodos exactos, se ha abordado a través de métodos heurísticos el problema de secuenciación de vehículos, estos métodos han demostrado proporcionar soluciones factibles. Sin embargo, en estos modelos la búsqueda de soluciones se realiza en forma no estructurada, pasando por alto las relaciones causales dentro del sistema.

1.1 Objetivos

Este trabajo tiene por objetivo abordar el problema de secuenciación considerando las relaciones causales inherentes a la arquitectura del problema, para establecer una búsqueda basada en una representación estructurada del conocimiento que permita al sistema productivo completar los ciclos de manufactura de las líneas multiproducto de manera eficiente. Para ello, es necesario un formalismo de modelado que proporcione el análisis de las relaciones causales.

Los objetivos específicos se estructuran de la siguiente forma:

- Modelar el problema de secuenciación de la industria de automóviles en un formalismo que permita determinar las relaciones causa - efecto que inciden en la toma de decisiones operacional.
- Proporcionar una herramienta y su formalización en un algoritmo matemático que se beneficie de los errores en las secuencias para acelerar la búsqueda a una secuencia factible.
- Desarrollar una herramienta computacional con el objetivo de experimentar la heurística propuesta.

1.2 Alcances y Limitaciones

El desarrollo del modelo para el problema de secuenciación será desarrollado bajo el formalismo de Redes de Petri Coloreadas, formalismo que como se verá en el capítulo 3 ha demostrado contar con los elementos para el modelado causal de sistemas logísticos. Una vez desarrollado el modelo se formalizará su descripción a través de un lenguaje matemático que proporcionará los elementos suficientes para desarrollar una herramienta computacional que permitirá conocer a través de simulaciones una solución factible al problema de secuencia.

Se ha elegido el sistema de producción de la industria de automóviles para ejemplificar la problemática y mostrar los diferentes modelos y formas de abordar el problema a lo largo de los años.

En ésta investigación el problema se ha acotado a la búsqueda de la mejor configuración para una línea de producción multiproducto de automóviles, dejando de lado los problemas de cambio de pintura definidos en el problema original.

Así pues no se pretende resolver a través de métodos exactos el problema de secuencia, si no reparar a través de la simulación errores potenciales de una secuencia introduciendo las permutaciones necesarias con el objetivo de que el sistema productivo no interrumpa actividades.

1.3 Organización del Trabajo

El presente trabajo presenta una propuesta para el problema de secuenciación de automóviles, en la primera parte se introducen los concepto básicos de logística y producción, así como el estado del arte y las diferentes metodologías propuestas a lo largo del tiempo.

En el capítulo 2, se encuentran los conceptos básicos de las Redes de Petri en general, formalismo que da lugar al modelaje del problema de estudio. Se describe la representación de los sistemas logísticos en Redes de Petri Coloreadas, en el capítulo 3.

El capítulo 4, expone el algoritmo, que aporta la propuesta de solución del problema de intercambio; este algoritmo matemático, describe el sistema y representa los problemas que se enfrenta un

sistema de producción, acotándolo al problema de la ruptura de stock debido a configuraciones en la secuencia inicial.

Un ejemplo de aplicación es presentado en el capítulo 5, a través del desarrollo de una aplicación para la automatización del algoritmo así como algunos resultados obtenidos para problemas de diferentes tamaños, finalmente en el capítulo 6 se presentan las conclusiones y las futuras líneas de investigación.

Capítulo 2

MARCO TEÓRICO

En este apartado se introducen los conceptos, definiciones y aspectos más relevantes de las diferentes técnicas que han sido desarrolladas para la resolver el problema de secuenciación.

2.1. Logística

De acuerdo con la European Logistics Association, [ELA] logística “es la organización, planificación, control y ejecución del flujo de materiales, desde el desarrollo y aprovisionamiento, pasando a través de la producción y hasta la distribución al cliente final, persiguiendo el objetivo de satisfacer los requerimientos del mercado al mínimo costo y con la mínima inversión de capital” [Sablon, 2009].

En logística se consideran tres sistemas fundamentales: aprovisionamiento, producción y distribución (figura 2.1), concebidos de forma integral y enfocada hacia la satisfacción del cliente:

Sistema de aprovisionamiento: Comprende todas las actividades que permiten que se muevan desde los puntos proveedores hasta la empresa.

Sistema de producción: Asume la responsabilidad de la transformación de los materiales adquiridos mediante el proceso de aprovisionamiento en productos para su posterior distribución.

Sistema de distribución: Mediante la distribución el cliente recibe los productos que se han obtenido en la producción.

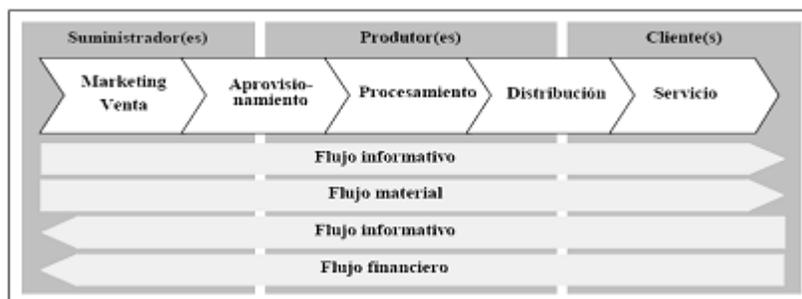


Figura 2.1 Sistemas fundamentales en logística.

2.2. Sistemas de Producción

Un sistema es una colección de objetos o entidades que interactúan entre sí para alcanzar un cierto objetivo [Piera, 2003], así un sistema de producción puede definirse como un conjunto de actividades y recursos orientados a la transformación de insumos en productos.

Dentro de la industria manufacturera, los sistemas de producción son los encargados de la elaboración de bienes, en donde los administradores de operaciones deben tomar decisiones que inciden sobre el funcionamiento del sistema.

El sistema inicia su acción con el análisis y cuantificación de las operaciones a realizar para satisfacer la demanda de producción por lo que es necesario buscar el cumplimiento eficaz de los objetivos de la organización, mediante la optimización de las partes para alcanzar una relación insumo/producto eficiente, logrando como resultado parte del alcance de las metas que se ha fijado la empresa.

Basado en lo anterior, la gestión de un sistema de producción debería contener las siguientes etapas:

- Un modelo que especifique las diferentes etapas del proceso productivo.
- Las relaciones causales entre los elementos de producción.
- Una estrategia de control que especifica la secuencia en que deben ser procesadas las órdenes, y la forma de resolver los conflictos debido a diferencias entre el modelo y el sistema real.
- Un mecanismo que se encarga de aplicar las soluciones para los problemas encontrados.

2.3. Modelos de Toma de Decisiones

Los sistemas de producción son desarrollados con base a la toma de decisiones sobre insumos, transporte, almacenaje, y administración de stocks entre otros, ejemplo de ello se puede observar cuando se están considerando recursos compartidos: la decisión de la operación a realizar en función del recurso, la elección de las piezas a procesar dentro de la línea de espera; y además el efecto que causa en el resto de los recursos compartidos con los que interactúa directa o indirectamente. Así es posible utilizar modelos de toma de decisiones para representar un sistema productivo en términos matemáticos.

Un modelo de toma de decisiones se expresa en términos de medidas del desempeño, limitantes y variables de decisión. El propósito de dicho modelo es encontrar los valores óptimos o satisfactorios para las variables de decisión que puedan mejorar el desempeño de los sistemas dentro de las restricciones aplicables.

Los primeros enfoques llegan alrededor de 1915 con FW Harris al desarrollar modelos para la administración de inventario basada en la cantidad económica del pedido. En 1931, Shewhart desarrollo los modelos de decisión cuantitativa para utilizarse en el trabajo de control estadístico de

calidad. En 1947 George Dantzing desarrolló el método simplex para la programación lineal, lo cual hizo posible la solución de una clase completa de modelos matemáticos. En la década de los 50 el desarrollo de los modelos de simulación por computadora contribuyó mucho al estudio y al análisis de operaciones. Y desde 1950 se ha expandido ampliamente el uso de varios modelos de toma de decisiones en las operaciones.

La Investigación de Operaciones (IO) formalizada en los años 50's se define como disciplina de aplicación de métodos analíticos avanzados para ayudar en la toma de mejores decisiones por el Institute for Operations Research and the Management Sciences, a partir de entonces se presenta como herramienta para la toma de decisiones, tomando en consideración que optimizar implica tomar una decisión para determinar las condiciones de operación del sistema que proporcione los mejores valores para el funcionamiento del mismo.

Sin embargo, muchos problemas de optimización no pueden ser abordados por métodos matemáticos, como la IO, ya que existe la gran dificultad en generar un modelo que represente el sistema real o se trate de problemas difíciles de optimización combinatoria, los cuales resuelven instancias de problemas que se creen ser difíciles en general, explorando el espacio de soluciones (usualmente grande) para estas instancias, como los problemas de planificación de la producción, redes, transporte y logística en general.

2.4. Secuencia y Programación de Tareas

En los sistemas de producción se pueden presentar varios problemas; uno a destacar es el relativo al equilibrio de capacidad entre las distintas unidades de producción, caso que se produce cuando una máquina situada en una cadena de producción o línea de montaje, tiene una carga de trabajo muy superior a la de las restantes; es entonces cuando se producen los denominados "cuellos de botella", que son estrangulamientos del proceso de producción que provocan la espera de piezas entre las células de trabajo, con las consiguientes pérdidas de productividad por la inactividad de las restantes estaciones de trabajo. Este problema exige un detallado estudio del tiempo de trabajo necesario para la ejecución de cada una de las operaciones y sus relaciones, a fin de conseguir un correcto equilibrio de las líneas de producción. De esta forma en un sistema de producción que manufactura más de un producto, el desequilibrio de la capacidad puede venir relacionado, con el orden en que las piezas de los diferentes productos son procesadas en una misma estación.

En general el orden en el que cualquier tipo de tarea será realizada, determina algunos de los parámetros influyentes en el costo total de la producción, en el caso particular de la producción en línea para diferentes tipos de productos, la inactividad de las estaciones de trabajo, debido a los stocks mal generados, puede provocar a lo largo de la jornada, un número menor de piezas procesadas.

Esto conduce en forma directa al problema de determinar cuál es el orden más adecuado para llevar a cabo las tareas con el objetivo de optimizar la producción. Se trata de un problema de secuenciación o scheduling que se presenta en forma habitual en la programación de operaciones a corto plazo en entornos industriales o manufactureros [Heizer y Render, 2001].

Este caso de problema de secuenciación es de tipo combinatorio [Halevi G., 2003], es importante destacar que resolver problemas combinatorios es encontrar la solución óptima dentro de un conjunto finito de alternativas, asumiendo que la calidad de la solución es cuantificable y

comparable con cualquier otra solución; sin embargo, en muchos problemas, dependiendo de su complejidad, llegar a la solución óptima requiere un proceso excesivamente largo.

Para resolver el problema de secuencia tamaño n , es necesario analizar todas las posibles configuraciones resultantes de las permutaciones de los elementos del conjunto proporcionado: $n!$,¹, por ejemplo considerando una línea de ensamble de automóviles donde se desee establecer la secuencia en la línea de producción de 3 vehículos, es decir, con $n=3$ el número de vehículos a ordenar, se tienen 6 posibles secuencias, como se muestra en la figura 2.2:

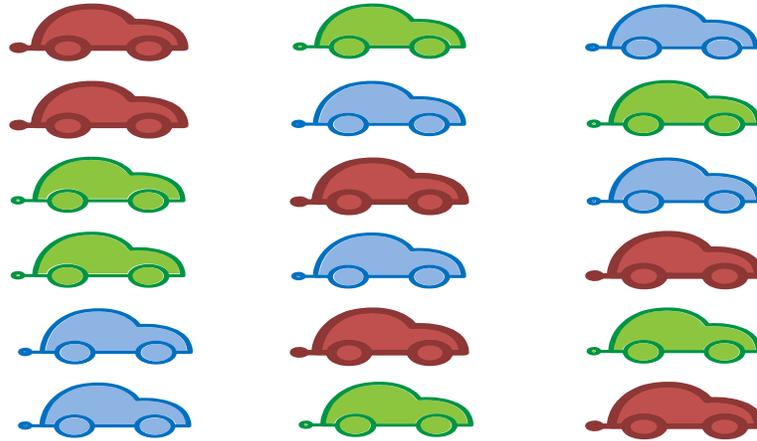


Figura 2.2 Ejemplo de configuración para 3 elementos.

La mayor parte de los problemas de optimización combinatoria pueden ser clasificados como NP-completo, una clase de problemas para los que no existe un algoritmo de tiempo polinomial que pueda resolverlos de manera óptima. Esto ha llevado a la exploración diversos métodos para abordar estos problemas, la mayoría de estos métodos pueden ser clasificados como algoritmos “exactos” o “heurísticas” [Aarts y Lenstra, 1993].

Los algoritmos exactos son aquellos que producen una solución óptima empleando técnicas con objeto de reducir el espacio de búsqueda. Dentro de este grupo se encuentran aquellos basados en los métodos de ramificación y acotamiento, planos cortantes, y programación lógica de restricciones; dichos algoritmos son eficientes para problemas de tamaño moderado [Ignizio y Cavalier, 1994; Moraga, Whitehouse, Depuy, 2003]. Inicialmente es posible resolver problemas de cualquier tamaño; sin embargo, en la práctica debido al gran número de soluciones posibles, estos métodos no son utilizados para problemas grandes, es decir garantizan encontrar la solución óptima, pero si el problema es complejo, el tiempo empleado en encontrar y garantizar la solución deseada no es viable.

Por otro lado las heurísticas no garantizan encontrar una solución óptima; no obstante, proporcionan una solución factible en un período de tiempo permisivo; éstas heurísticas se pueden clasificar en problemas combinatorios de construcción y de mejoramiento; la heurística de construcción produce una solución factible como resultado final, en tanto que una heurística de mejoramiento comienza a partir de una solución factible y luego la mejora [Reinelt, 1994].

¹ $n = 20 \Rightarrow n! = 2.4329E + 18$

El desarrollo de metaheurísticas (también llamadas heurísticas modernas), junto al avance de las tecnologías de información ha permitido la solución de problemas combinatorios complejos [Yu, 1998]. En general, estas metaheurísticas toman inicialmente una solución factible, posteriormente ésta solución es mejorada empleando estructuras generales; como ejemplo de lo anterior se encuentra: Simulated Annealing (SA), Algoritmos Genéticos (GA), Búsqueda Tabú (TS), y Redes Neuronales (NN). La característica común de estos enfoques es el uso de mecanismos para evadir óptimos locales, uno de estos mecanismos es el uso de aleatoriedad; puede ser demostrado que aplicando cierto grado de aleatoriedad y combinando diferentes heurísticas es posible obtener mejores soluciones para un problema de optimización combinatoria particular [Whitehouse y DePuy, 2001] .

2.5. Estado del arte. Problema de secuenciación de la industria del automóvil

El problema de secuencia de vehículos ha sido abordado desde diferentes métodos, que se abordarán más adelante, este problema fue descrito por [Parello 1986] y consiste en la planificación de vehículos a lo largo de una línea de ensamble para la instalación de diferentes opciones.

La capacidad de las estaciones donde se instalan estas opciones (por ejemplo radio, aire acondicionado, quemacocos) es limitada. Las estaciones están diseñadas para operar cierto porcentaje de automóviles que pasan a lo largo de la línea de ensamble y los vehículos que requieren la opción, deben estar secuenciados de tal forma que la capacidad de cada estación nunca sea excedida.

Los requerimientos pueden ser formalizados por una tasa de restricción p/q : cualquier sub-secuencia de q vehículos comprende al menos p vehículos que requieren esta opción.

El problema de secuenciación de vehículos es definido por la tupla $(V,O,p,,q,r)$, donde:

$V=\{v_1, \dots v_n\}$ Conjunto de vehículos a ser producidos.

$O=\{o_1, \dots o_m\}$ Conjunto de diferentes opciones.

$p,q:O \rightarrow \mathbf{N}$ Define la restricción de capacidad asociada a cada opción $o_i \in O$.

$r:V \times O \rightarrow \{0,1\}$ Define las opciones de requerimiento, i.e. para cada vehículo $v_j \in V$ y para cada opción $o_i \in O$, $r_{ij}=1$ si o_i debe ser instalado en v_j y $r_{ij}=0$ en otro caso.

Aquí dos diferentes automóviles pueden requerir la misma configuración, i.e. las mismas opciones; entonces todos los vehículos que requieran la misma configuración de opciones serán parte de la misma clase de automóviles: existen k diferentes clases de automóviles tal que V se encuentra particionado en subconjuntos, $V=V_1 \cup V_2 \cup \dots V_k$.

Se ha planteado y resuelto el problema a través de métodos de optimización, donde se establece la función objetivo como la minimización del costo en función a la violación de restricciones. En la formulación original del problema [Parello, Kabat, Wos, 1986], el costo depende de las opciones

que son afectadas por la violación de las restricciones; el número de vehículos que exceden la capacidad, y la ubicación de éstos en la secuencia, a partir de esta definición diferentes autores han establecidos funciones objetivo que establecen en general la minimización de la violación de restricciones.

Este problema de decisión se encuentra clasificado como NP Hard, donde el espacio de estados se encuentra dado de forma factorial dependiendo del número de clases de automóviles que se esté involucrando en el problema. El problema de secuencia de automóviles se ha estudiado principalmente como un problema de optimización. Muchas técnicas tales como ramificación y acotamiento, búsqueda del tabú, colonia de la hormiga y otros acercamientos heurísticos se han desarrollado para solucionarlo.

La búsqueda de métodos para solucionar el problema de secuencia de automóviles dio origen al desafío ROADEF'2005; donde diferentes artículos fueron publicados para solucionar este problema [Solnon, 2000]. El reto ROADEF es organizado cada dos años por la Sociedad Francesa de Investigación de Operaciones y Análisis de Decisiones con el fin de permitir que los socios industriales acrediten los progresos recientes en el campo de la investigación de operaciones y del análisis de decisión, así como los investigadores traten un problema real, y a menudo complejo, que ha surgido en la industria. En 2005, el tema para este desafío fue propuesto por la firma Automotriz RENAULT orientado al problema de secuenciación de automóviles. Este problema implica el programar de los vehículos a lo largo de una línea de ensamble con restricciones estrictas y permisivas.

El problema que se ha planteado por la compañía Renault, se describe a continuación [Solnon, C., Cung, V-D., Nguyen A., Artigues C 2008]:

Las órdenes de los clientes son enviadas a las plantas en tiempo real. Las tareas diarias de cada planta es (1) asignar a cada vehículo un periodo de manufactura, tomando en cuenta las restricciones de capacidad de la línea de ensamble y las fechas de vencimiento. La siguiente tarea (2) es colocar los vehículos en la línea de producción, satisfaciendo de la mejor forma posible los requerimientos de producción de workshops: Body (Carrocería), pintura y ensamble. La parte de Carrocería no es considerada crítica en la producción de vehículos, por lo que el problema presentado por RENAULT se centra en las células de Pintura y Ensamble (figura 2.3).

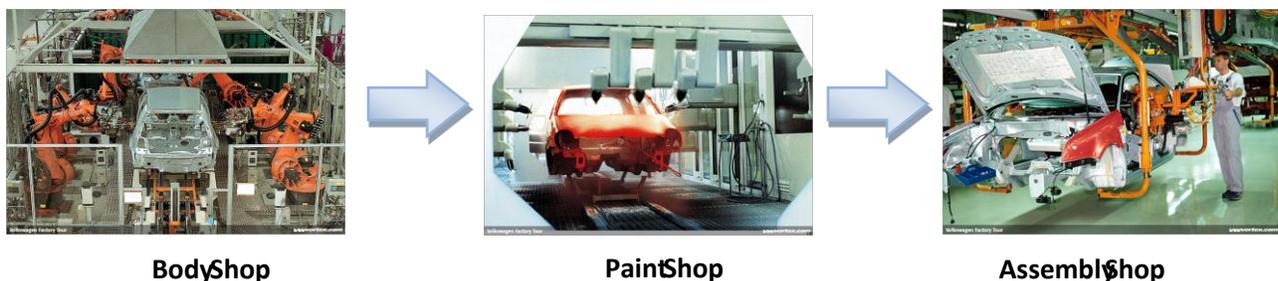


Figura 2.3 Ruta de producción crítica.

Requerimientos de Pintura. El principal objetivo es minimizar el consumo de solvente que es usado al limpiar la maquinaria al cambio de color; esto es alcanzado si se coloca en la secuencia vehículos que deben ser pintados del mismo color; sin embargo, aun utilizándose el

mismo color, se debe limpiar el equipo periódicamente. Esta es una restricción fuerte y no debe ser violada.

Requerimientos de Ensamble. Aquí el objetivo es mantener un balance de las diferentes unidades de trabajo en las líneas de ensamble. Los vehículos que requieren operaciones complejas deben estar lo suficientemente distantes dentro de la secuencia, es decir, la densidad, de esos vehículos difíciles, debe estar delimitada para prevenir sobrecargas de trabajo en las unidades que montan estos vehículos.

A continuación se presentan los diferentes métodos que han sido utilizados para resolver el problema de secuenciación de automóviles:

Programación de Restricciones (CP)

Es una herramienta utilizada para resolver Problemas de Satisfacción de Restricciones (CSP), que son problemas con restricciones específicas sobre soluciones aceptables, donde una restricción es una relación entre variables o incógnitas y cada variable toma un valor de un dominio dado [Tsang, E.P.K. 1993]. Estos problemas son resueltos con un CP lenguaje, donde se debe especificar las variables y restricciones, el proceso de solución está a cargo de los algoritmos genéricos que se encuentran integrados dentro del CP lenguaje. Estos algoritmos generalmente están basados en exploraciones sistemáticas del espacio de estados hasta encontrar una solución o hasta probar la inexistencia de la solución. Para reducir el espacio de estados, la aproximación es combinada con filtros que reducen el dominio de las variables con respecto a consistencias parciales como Arco-Consistencia [Barber, F., Salido, MA 2003].

Esta técnica ha mostrado su eficacia para resolver problemas de secuenciación pequeños, pero no es competitivo aplicado a problemas grandes. Se ha establecido variaciones para probar el proceso de solución de la programación de restricciones; sin embargo, estos filtros o modificaciones a algoritmos no logran resolver el problema en un tiempo razonable [Smith, B.1996; Regin, J.-C. Puget, J.-F. 1997; Van Hoes, W.-J. Pesant, G. Rousseau, L.-M. 2006; Lee, J.H.M., Leung, H.F., Won, H.W. 1998].

Programación Entera

Se ha propuesto el modelado del problema de secuenciación de vehículos a través de la programación entera. El modelo se basa en variables binarias asociadas con la clase de auto y su posición en la secuencia. El problema de secuenciación es mezclado con un problema de programación que minimiza la suma de desviaciones dentro del periodo planeado; lo que deriva en un problema de secuenciación con restricciones duras de capacidad y un nivel de programación objetivo [Drexler, A. Kimms, A. 2001]. También se ha propuesto para el problema de secuenciación de automóviles la programación entera con restricciones suaves de capacidad [Gravel, M., Gagne, C., Price, W.L. 2005].

Método Ad-hoc

El problema de secuenciación y planificación ha sido abordado por métodos de ramificación y acotamiento, considerando solo la parte de la secuencia de automóviles, el método está basado en un esquema de ramificación a partir del estado de la secuencia de automóviles.

Un estado de la secuencia se encuentra asociado a cada nodo de la rama del árbol [Drexl, A., Kimms, A., Matthiessen, L. 2006]. Además la secuencia parcial correspondiente al nodo del estado se encuentra definida por una matriz $m_{(i,j)} \in \{0,1,-1\}$ donde $m_{ij} = 0$ si la opción i está presente en la posición j , $m_{ij} = 1$ si la opción i puede ser planeada en la posición j ; $m_{ij} = -1$ si la opción i no puede ser planeada en la posición j , debido a una rama de decisión o porque se pudiera violar una restricción. Las condiciones necesarias y el estudio de mecanismos basados en los estados de la secuencia de automóviles son entonces usados para podar la búsqueda.

No sólo se han utilizado aproximaciones exactas, también se han empleado diferentes métodos heurísticos para la resolución del problema:

Greedy Approaches

Son también llamados “algoritmos de búsqueda glotona”, en este método se puede construir una secuencia de forma “amplia” comenzando desde una secuencia vacía e iterativamente ir agregando nuevos vehículos al final. Se han propuesto y comparado seis diferentes tipos de algoritmos glotones para el problema de secuenciación de automóviles [Gottlieb, J., Puchta, M., Solnon, C. 2003]; entre ellas la que ha dado mejores resultados se basa en la suma dinámica de las tasas de utilización, i.e. cada paso agrega un auto que maximiza la suma de las tasas de utilización de las opciones requeridas. Estas tasas de utilización se actualizan cada vez que un auto se agrega al final de la secuencia. Este tipo de construcción cuando esta combinada con una distribución aleatoria y múltiples inicios puede resolver rápidamente los casos que se plantean en CSPLib de Lee [Lee, J.H.M., Leung, H.F., Won, H.W. 1998].

Búsqueda Local

El objetivo de la búsqueda local es mejorar una secuencia a través de la exploración en una vecindad, es decir, el conjunto de secuencias que puede ser obtenido de la secuencia actual por transformaciones elementarias llamadas “movimientos”. Dada una secuencia inicial, el espacio de búsqueda es explorado de vecindad en vecindad hasta encontrar una secuencia óptima o hasta que el número máximo de “movimientos” se haya agotado. Este método ha sido utilizado para resolver el problema de secuenciación de automóviles [Puchta, M., Gottlieb, J., 2002; Gottlieb, J., Puchta, M., Solnon, C. 2003], y ha resultado ser efectivo en todos los casos [Michel, L., Van Hentenryck, P., 2002; Gottlieb, J., Puchta, M., Solnon, C. 2003; Neveu, B., Trombettoni, G., Glover, F., 2004; Perron, L., Shaw, P., 2004; Perron, L., Shaw, P., Furnon, V., 2004].

Las aproximaciones de la búsqueda local para resolver el problema de secuenciación de automóviles principalmente difieren con respecto a la forma en que se construye la secuencia inicial, la vecindad considerada en cada movimiento y la metaheurística considerada para elegir un movimiento entre vecindades.

Algoritmos Genéticos

La aproximación de algoritmos genéticos está inspirada en la evolución natural, explora el espacio de estados a través de la selección, cruza y mutación de operadores. La utilización de este método en el problema de secuenciación de automóviles [Warwick, T., Tsang, E., 1995] muestra que esta aproximación es capaz de resolver fácilmente el problema; sin embargo, con un alto porcentaje de

utilización, el número de éxitos decrece. Se ha propuesto un algoritmo computacional para resolver el problema práctico de secuenciación de Ford Motor [Company Cheng, J., Lu, Y., Puskorius, G., Bergeon, S., Xiao, J., 1999], en el que se propone un operador llamado cross-switching que genera una descendencia mediante el intercambio de vehículos que aparecen en una posición elegida en la secuencia de los padres.

Colonia de Hormigas

La idea básica de optimización de colonia de hormiga (ACO) por sus siglas en inglés [Dorigo, M., Stutzle, T., 2005] es de modelar el problema como la búsqueda de un camino de coste mínimo en un gráfico, y usar “hormigas artificiales” para búsqueda de caminos factibles. El comportamiento de las “hormigas artificiales” esta inspirados en el comportamiento real: las hormigas ponen rastros de feromona sobre los componentes del gráfico y así las hormigas siguientes eligen caminos de acuerdo a las probabilidades que dependen de la feromona que antes ha sido depositada por la colonia; los rastros de feromona que no han sido reforzado por el paso continuo de la colonia se evaporan. La comunicación indirecta, tiene por objeto dar información acerca de la calidad de la ruta, a fin de atraer a las hormigas, en las siguientes iteraciones.

En el algoritmo de ACO [Solnon, C., 2000], se establece feromonas sobre los pares de automóviles consecutivos con el fin de aprender de las subsecuencias “prometedoras”. Este primer algoritmo se ha mejorado [Gottlieb, J., Puchta, M., Solnon, C., 2003] por la integración de una heurística glotona, y ha sido experimentalmente comparado con la el enfoque de búsqueda local [Puchta, M., Gottlieb, J., 2002], demostrando que la búsqueda local es ligeramente inferior al ACO en periodos de tiempo cortos de CPU mientras que para periodos más grandes la solución es comparable.

Las metodologías aquí expuestas han abordado el problema de secuencia, bajo el diseño inicial de la optimización, planteándolo a través de la función objetivo, con diferentes variantes y restricciones, a partir de éste planteamiento se propone resolver el problema objeto de ésta investigación, sin embargo se considera necesario modelar el problema bajo esquemas de relaciones causa-efecto, por lo que si bien las metodologías han resuelto el problema, bajo ciertos parámetros, queda aún estudiar el problema desde el punto de vista sistémico.

En el siguiente apartado se describen las bases de las Redes de Petri, formalismo propuesto para integrar los efectos causales de sistemas logísticos como es el problema de secuencia en la industria automotriz.

Capítulo 3

REDES DE PETRI

Dentro de las diferentes metodologías que han sido usadas tradicionalmente para dar respuesta a los problemas de planeación, las técnicas de modelado y simulación han probado ser útiles en su estrategia y diseño; sin embargo, las limitaciones aparecen cuando se busca una solución a un problema NP-completo ya que sólo un limitado número de escenarios puede ser evaluado en un tiempo razonable.

Los problemas que utilizan técnicas de optimización son diseñados en términos de una función objetivo y un conjunto de restricciones, requerimientos y límites, no obstante, existen muchos problemas que presentan un comportamiento no lineal, relaciones combinatorias e incertidumbres que no pueden ser fácilmente formalizado como un conjunto de variables y una función objetivo. Así, los métodos de optimización se pueden utilizar en la planeación operacional de sistemas estocásticos, dinámicos y sincronizados cuando el problema es simplificado [Piera MA, Guasch A 2003]. Desafortunadamente la solución óptima de un problema simplificado puede estar lejos del óptimo del problema real o no satisfacer sus requerimientos.

Las Redes de Petri Coloreadas han demostrado ser una herramienta para el modelado de sistemas complejos debido a la representación de estructuras estáticas y dinámicas, la disponibilidad de técnicas de análisis matemático así como su naturaleza gráfica [Silva 1989, Zimmerman 1996, Jensen 1997].

La potencia en cuanto a herramienta de modelado que ofrece el formalismo de las Redes de Petri respecto a otros formalismos, radica en la descripción de la relación estado-evento. Estas redes permiten presentar los cuatro conceptos básicos de modelado: dependencia causal (secuencias de eventos), nodos de decisión, concurrencia y sincronismo, que son factores claves cuando se requiere optimizar el funcionamiento de sistemas logísticos o de manufactura.

3.1 Descripción de una Red de Petri (RdP)

Una red de Petri (RdP) es un grafo orientado en el que intervienen dos clases de nodos, los lugares (representados por circunferencias) y las transiciones (representadas por segmentos rectilíneos), unidos alternativamente por arcos. Un arco une un lugar con una transición, o viceversa, pero nunca dos transiciones o dos lugares.

Un lugar puede contener un número positivo o nulo de marcas. Una marca se representa por un punto en el interior del círculo correspondiente al lugar. El conjunto de marcas asociadas en un instante dado a cada uno de los lugares, constituye un marcado de la RdP, todos estos elementos se muestran en la figura 3.1.

Para la descripción funcional de sistemas, a los lugares se les asocian la descripción tanto de colas de espera como condiciones sobre el estado en que se encuentran los elementos o recursos que forman parte del sistema; y a las transiciones les son asociadas eventos (funciones lógicas de las variables de entrada del sistema), acciones o actividades que aparecen en la dinámica del sistema.

En una RdP el estado está representado por el marcado. Como el marcado puede contener varios lugares marcados simultáneamente, el estado estará definido por un conjunto de subestados (estados locales o parciales) del sistema.

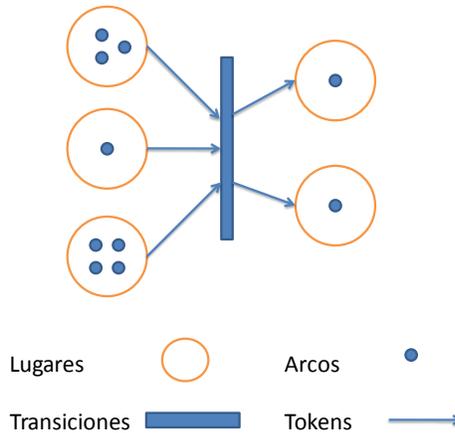


Figura 3.1 Representación de una Red de Petri.

3.2 Definición Formal de las Redes de Petri (RdP)

Matemáticamente una RdP se define como una tupla compuesta de cinco elementos [Proth y Xie 1996; Guasch 2005]:

$RP = (P, T, A, W, M_0)$, donde:

- $P = \{P_1, P_2, P_3, \dots, P_{np}\}$

Conjunto finito no vacío de nodos tipo lugar (np es el número de nodos lugar de la RdP).
- $T = \{T_1, T_2, T_3, \dots, T_{nt}\}$

Conjunto finito no vacío de nodos tipo transición (nt es el número de nodos transición de la RdP).
- $A = \{A_1, A_2, A_3, \dots, A_{na}\}$

Conjunto finito no vacío de arcos (na es el número de arcos de la RdP).
- $A \subset (P \times T) \cup (T \times P)$

Subconjunto del producto cartesiano de los conjuntos P y T.
- $W: A_i \rightarrow \{1, 2, 3, \dots\}$

Peso asociado a cada arco $\forall i = 1, 2, \dots, na$.
- $M_0 = [p_1, p_2, p_3, \dots, p_{np}]$

Marcado inicial donde p_j es el número de marcas en el j-ésimo nodo tipo lugar.

El estado de un sistema después de la ocurrencia de un evento, queda determinado por el número de marcas en cada nodo tipo lugar y puede describirse matemáticamente por el vector:

$$M_1 = [p_1, p_2, p_3, \dots, p_{np}] \quad \forall i = 1, 2, \dots, k$$

Donde k representa el número total de marcados (estados) del sistema.

3.3 Dinámica y representación de las Redes de Petri (RdP)

Para representar y analizar una RdP se define la función $E(T_j)$ que representa el conjunto de lugares de entrada a la transición T_j , análogamente se define $S(T_j)$ para identificar los lugares de salida:

$$\begin{aligned} E(T_j) &= \{P_i \in P, (P_i, T_j) \in A\} \\ S(T_j) &= \{P_i \in P, (T_j, P_i) \in A\} \end{aligned}$$

Las reglas para simular el comportamiento de una RdP son:

- La transición T_i esta activada si cada uno de los nodos P_j tipo lugar conectados a la entrada contiene al menos $W(P_j, T_i)$ marcas. $W(P_j, T_i)$ representa el peso del arco que une al nodo P_j con la transición T_i , si en un arco no aparece el peso, se da por entendido que el peso es uno. De lo anterior decimos que una transición T_i , esta activada si:

$$M(P_j) \geq W(P_j, T_i) \quad \forall P_j \in E(T_i)$$

Donde $M(P_j)$ es el número de marcas en el nodo tipo lugar P_j .

- Una transición activada puede dispararse en cualquier momento.
- Como resultado de disparar una transición activada se eliminan $W(P_j, T_i)$ marcas de cada nodo P_j a la entrada de T_i y se añaden $W(T_i, P_k)$ marcas a cada nodo P_k de salida de T_i . Donde $W(T_i, P_k)$ corresponde al peso del arco que une a la transición T_i con el nodo P_k .

Es decir una transición esta deshabilitada cuando hay menos marcas en cualquiera de los nodos lugar de entrada de la transición de los que señalan los respectivos arcos de entrada a dicha transición, en caso contrario la transición está habilitada.

Cuando una transición se encuentra habilitada, ésta puede estar disparada. Este efecto se ve reflejado en la disminución de marcas de los nodos lugar de entrada de acuerdo al peso del arco y el incremento, de acuerdo al peso del arco, en el nodo lugar de salida como se muestra en la figura 3.2.

Cuando existen una o más transiciones habilitadas, éstas pueden ser disparadas; si además existen suficientes marcas en los nodos lugar de entrada de cada transición tal que cada una de éstas pueda obtener sus propias marcas sin tener que compartir recursos con otras transiciones,

se dice que las transiciones están concurrentemente habilitadas, i.e. las transiciones pueden ser disparadas al mismo tiempo (en paralelo).

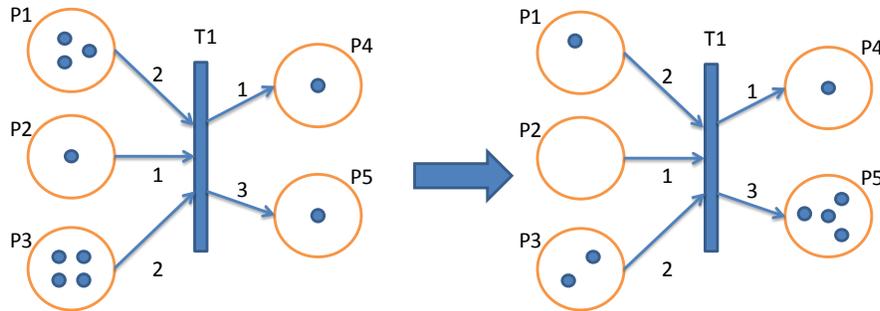


Figura 3.2 Dinámica de una RdP.

Una de las ventajas más grandes es la representación de los mecanismos de los procesos orientados a eventos discretos los cuales se caracterizan por que las propiedades de interés del sistema cambian únicamente en una secuencia de instantes de tiempo:

- Ejecución secuencial. Refleja la dependencia causal entre eventos. La transición T2 sólo puede ser disparada después de activarse la transición, figura 3.3 a.
- Toma de decisiones. Las transiciones T1, T2 y T3 de la figura 3.3 b están habilitadas para su disparo; sin embargo, el disparo de cualquiera de ellas supone la deshabilitación del resto.
- Concurrencia. La figura 3.3 c, muestra un ejemplo de transiciones concurrentes.

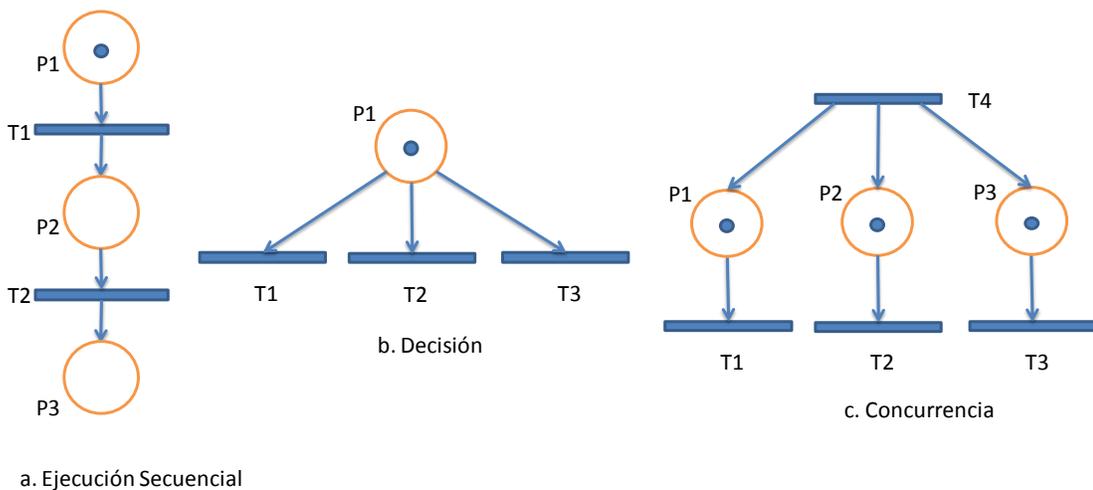


Figura 3.3 Representación de procesos sencillos (1).

- Sincronización. La transición T1 no se puede disparar hasta que no se tenga una marca en el lugar P1. Esta estructura es empleada cuando se está en espera de un recurso para proseguir, o bien en procesos de ensamblado donde se requieren todas las piezas para realizar una operación. Representación 3.4 d.
- Agrupación. Las marcas llegan por diferentes caminos y se agrupan en un lugar. Un ejemplo de esta estructura son los pacientes que han seguido diferentes caminos para acudir a consulta médica. Representación 3.4 e.
- Inhibición. Representada con un arco de inhibición, que no permite en este caso la transición T1 cuando P1 tiene una o más marcas, su efecto es el contrario al de los arcos vistos hasta este momento. Suele emplearse para priorizar transiciones frente a otras. Representación 3.4 f.

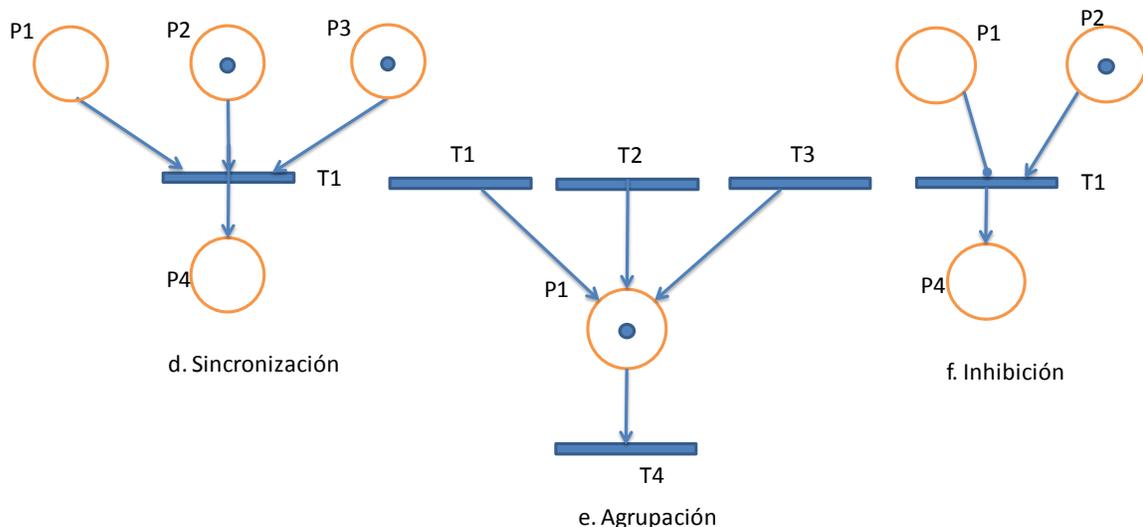


Figura 3.4 Representación de procesos sencillos (2).

3.4 Redes de Petri Coloreadas (RPC)

Las Redes de Petri Coloreadas permiten construir modelos compactos y paramétricos que requerirían estructuras con un número elevado de componentes si fueran desarrolladas con el formalismo de las RdP. Los modelos de RdP de sistemas logísticos complejos suelen estar compuestos de subredes similares, por lo que la representación compacta, en la que el comportamiento de tales subredes es formalizado por una sola red, facilita considerablemente la evaluación de las posibles configuraciones del sistema en estudio.

En su gran mayoría los sistemas logísticos requieren la especificación de características (atributos) para describir las entidades que fluyen a través del sistema, por lo tanto es necesario un formalismo que aporte a la descripción del flujo de tareas las características de cada entidad, refiriéndonos a las entidades como son objetos concretos o abstractos que presentan interés para el sistema y sobre los que se recoge información que será representada en el mismo, permitiendo especificar la priorización de las entidades en una cola de espera; cómo un evento puede afectar a

los valores de los atributos de la entidad procesada; la activación de eventos en función de los atributos de cada entidad; la duración de una actividad en función de los atributos de las entidades involucradas; el flujo de entidades y lo que ocurre en cada una de ellas conforme fluye a través de una secuencia de subsistemas físicos; Los cambios de estado del sistema y la secuencia de eventos que deberían aparecer para que una entidad finalice un tipo de procesamiento.

Las Redes de Petri Coloreadas conllevan un nivel de abstracción en la etapa del modelado gracias al uso de colores, los cuales representan los atributos de las entidades.

Se presenta el modelo de un proceso en paralelo de las RPC, mostrando su capacidad de simplificación en la figura 3.5:

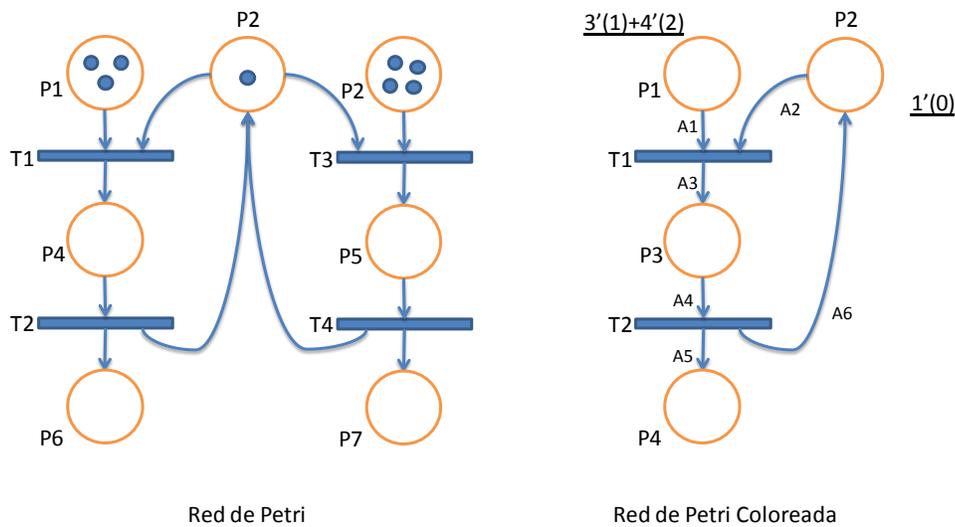


Figura 3.5 Simplificación en el diseño de un sistema en RdPC.

Se trata un sistema que procesará dos tipos de piezas en una estación de trabajo, los procesos son idénticos, no obstante, para la RdP es necesario dos circuitos, cada uno para un tipo de pieza diferente, cada pieza espera en el lugar P1 (3 del tipo 1), y P2 (4 del tipo 2) a que este la estación de trabajo (una estación de trabajo representada por una marca) P2 se encuentre desocupada para iniciar el proceso T1 para las piezas del tipo 1 y T3, para las piezas del tipo 3. El proceso se lleva a cabo en los lugares P4 y P5, Las transiciones T2 y T4 indican el fin del proceso, devolviendo el estado de libre a la estación de trabajo y estableciendo un stock (lugar P6, P7) para cada tipo de pieza.

Para transformar la RdP en una RPC, se formalizan los dos procesos mediante un único nodo lugar donde se diferencian el tipo de elemento que se está procesando y el estado de la máquina. De esta forma los nodos P1 y P3 se fusionan en un único nodo que alojarán las marcas de tipo 1 y tipo 2. En las RPC es necesario definir que colores deben representarse, una posible asignación es la fusión de los nodos lugar que describen el mismo recurso (stock delante de las máquinas, color x), por otro lado la identificación de la máquina que está procesando la pieza (color y) para el nodo

lugar P2; y por último un tercer color R que sea el producto cartesiano de los otros dos al nodo lugar P3:

$\Sigma=\{x,y,R\}$ donde

x:	tipo de pieza que se debe procesar (1,2)
y:	estado de la estación de trabajo (0,libre; 1, ocupada)
R:	producto cartesiano de x por y; da la información del estado de la máquina y del tipo de pieza que se procesa

3.5 Descripción de una Red de Petri Coloreada (RPC)

La diferencia entre una RdP y una RPC es la capacidad de la RPC de asociar a cada entidad (marca) un conjunto de valores denominado color de la marca (token color).

Para un nodo lugar dado todas las marcas deben tener colores del mismo tipo especificado. Este tipo se denomina conjunto de color (set color) del nodo lugar.

En las RPC además de los elementos básicos de las RdP, se anexan los conceptos de color y conjunto de color entre otros elementos:

Conjuntos de Color. Cada nodo lugar sólo puede tener marcas que pertenezcan al mismo tipo, el cual es conocido como conjunto color del nodo lugar. Gráficamente se representa con el nombre del conjunto color a un lado del nodo lugar.

Expresiones de inicialización. Indican el número inicial de marcas y el color en cada nodo lugar. Los colores de las marcas se especifican a través de una expresión subrayada junto al nodo lugar: $\underline{n'(c_1, c_2, c_3, \dots, c_k)}$ donde n es el número de marcas con los valores de color descritos dentro del paréntesis y ci representa el valor de cada componente de color.

Cuando los valores de los colores de las marcas no son iguales para todos los elementos del nodo lugar, se representa con “+” entre cada tipo: $\underline{n_1'(c_1, c_2, c_3, \dots, c_k) + n_2'(c_1, c_2, c_3, \dots, c_k)}$.

Estado Inicial M_0 . Se determina evaluando las expresiones de inicialización asociadas a cada nodo lugar, así como los valores de los colores de las marcas.

Expresiones de Arco. Los colores de las marcas son revisados en las transiciones, lo cual permitirá habilitarlas en función del número de marcas en los nodos lugar conectados a la entrada de la transición y en función de los valores de color de las marcas disponibles en dicho nodo lugar, al mismo tiempo que permite modelar los efectos de cada transición definiendo nuevos colores para las marcas de salida.

Las expresiones de arco consisten en la formalización de restricciones entre los colores de las distintas marcas de los nodos lugar conectados a la entrada de la transición, para esto se utilizan variables que han sido instanciadas a los valores concretos de las marcas, y que obligan a una selección de marcas cuyos colores coincidan con los valores de las variables instanciadas.

Guardas. Los guardas son expresiones lógicas que imponen ciertos colores a los colores de las marcas que pueden ser elegidos para habilitar una transición; éstos son representados entre corchetes al lado de la transición.

Marcado. Representa la información mínima necesaria para pronosticar cuales son los posibles eventos que pueden producirse. Se describe mediante la especificación del número de marcas en cada nodo lugar y los valores de los colores en cada una de las marcas.

Las RPC permiten tanto los atributos como las características de los objetos que fluyen en el sistema, como las propiedades que deben de tener para que un cierto pueda suceder. Ejemplo de atributos asociados a una pieza son: tiempo de llegada al sistema, fecha de entrega al cliente, prioridad, color, forma, etc.

3.6 Definición Formal de las Redes de Petri Coloreadas (RPC)

Formalmente una RPC se define como una tupla [Jensen 1997, Guasch 2005]:

$RPC = (\Sigma, P, T, A, N, C, G, E, I)$ donde:

$\Sigma = \{C_1, C_2, C_3, \dots, C_{nc}\}$ Conjunto finitos no vacíos de colores (nc es el número de conjuntos color especificados para la RPC).

$P = \{P_1, P_2, P_3, \dots, P_{np}\}$ Conjunto finito no vacío de nodos tipo lugar (np es el número de nodos lugar de la RPC).

$T = \{T_1, T_2, T_3, \dots, T_{nt}\}$ Conjunto finito no vacío de nodos tipo transición (nt es el número de nodos transición de la RPC).

$A = \{A_1, A_2, A_3, \dots, A_{na}\}$ Conjunto finito no vacío de arcos (na es el número de arcos de la RPC).

$N: A \rightarrow (P \times T) \cup (T \times P)$ Función Nodo, que permite asociar a cada arco sus nodos terminales en pares ordenados, tal que

$$\forall A_i \in A \exists! P_j \in P \text{ y } T_k \in T : \\ N(A_i) = (P_j, T_k) \text{ y } N(A_j) = (T_k, P_j)$$

Donde el primer elemento del par ordenado corresponde al nodo origen y el segundo de elemento al nodo destino. Los nodos son de tipo diferente por lo tanto si un nodo es de tipo lugar, el otro es de tipo transición y viceversa.

$C: P \rightarrow \Sigma$ Función Color, permite especificar para cada nodo lugar, el tipo de entidades (marcas) que se pueden almacenar,

$$\forall P_i \in A \exists! C_q \in \Sigma : C(P_i) = C_q$$

$G: T \rightarrow Boolean$ Función guarda, permite asociar a cada nodo transición una expresión del tipo lógico:

$$\forall T_k \in T \exists! C_q \in \Sigma : \\ \text{tipo}(G(T_k)) = Boolean \text{ y } \text{tipo}(\text{variable}(G(T_k))) \subseteq \Sigma$$

$E: A \rightarrow C(P_i)$ Función expresión de arco, permite especificar el tipo de entidad (marca) del nodo lugar de entrada en una transición que debe elegirse de las marcas alojadas en dicho nodo para habilitar la transición:

$$\forall A_i \in A : \\ \text{tipo}(E(A_i)) = C(P_j) \text{ y } \text{tipo}(\text{variables}(E(A_i))) \subseteq \Sigma$$

donde P_j representa el nodo lugar de entrada o salida del arco A_i ,

Cuando la expresión E se encuentra asociada a un arco de salida de la transición, la expresión se utiliza para evaluar los nuevos valores de color de los atributos de las entidades (marcas) de salida.

$I: P \rightarrow C(P_i)$ Función de inicialización, permite especificar los valores de color de los atributos de las entidades (marcas) inicialmente almacenadas en un nodo lugar:

$$\forall P_j \in P : \\ \text{tipo}(I(P_j)) = C(P_j)$$

3.7 Dinámica y representación de las Redes de Petri Coloreadas (RPC)

De la misma forma que en las RdP, las expresiones de arco en RPC indican las condiciones necesarias para la activación de una transición; sin embargo, para las RPC es necesario que se incluya el color de las marcas que habilitaran una transición además de satisfacer la condición que expresa el guarda asociado a la transición. Se describe a continuación un ejemplo de la dinámica para las RPC tomando en cuenta un sistema descrito con la red, figura 3.6.

La red describe un sistema de producción que cuenta con tres estaciones de trabajo y tres tipos de pieza, en su correspondiente stock.

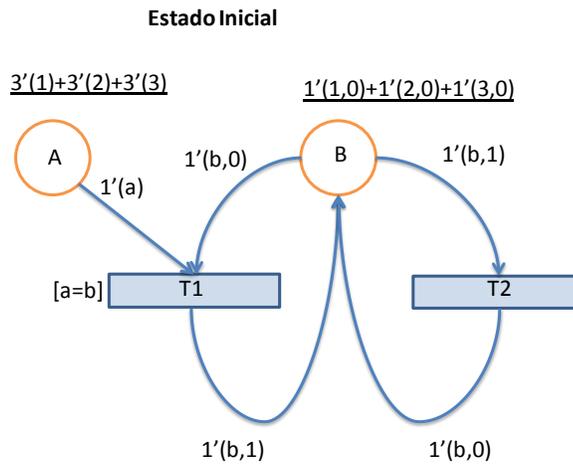


Figura 3.6 Estado Inicial en RPC.

El nodo lugar A, representa los tipos de piezas que entraran a proceso, el lugar B, es el estado en que se encuentran las estaciones de trabajo.

La transición T1 representa el evento que ocurre cuando una pieza del stock1 que alberga las piezas del tipo1, entra a proceso. La transición T2 representa el evento cuando la pieza ha sido procesada y sale de la estación de trabajo.

Para que la transición T1 este habilitada debe de existir al menos una marca del color a: expresión del arco $1'(a)$.

El lugar B, debe tener al menos una marca donde el primer componente de color sea b, y el segundo componente de color sea o: expresión de arco $1'(b,0)$.

El color de A debe ser igual al color del componente b: guarda $[a=b]$, i.e. el valor de la variable a debe ser igual al valor de la variable b.

Cumplidas estas condiciones, la transición está habilitada: Una pieza entra a proceso. Eligiendo la marca $1'(1)$ del nodo lugar A y la marca $1'(1,0)$ del nodo lugar B al disparar la transición T1, da origen al estado en el sistema: $M1=[2'(1)+3'(2)+3'(3), 1'(1,1)+1'(2,0)+1'(3,0)]$, es decir, hay dos piezas en el stock 1: $2'(1)$; 3 piezas esperando en el stock 2: $3'(2)$ y 3 piezas en el stock 3: $3'(3)$. La estación de trabajo1 se encuentra ocupada $1'(1,1)$ y las otras estaciones están desocupadas $1'(2,0)$; $1'(3,0)$. Cabe hacer notar, que la transición T1 puede activarse con otras marcas, esto significa que T1 se encuentra concurrentemente habilitada y el evento que representa puede ocurrir varias veces de manera simultánea. La dinámica del sistema se refleja en la figura 3.7.

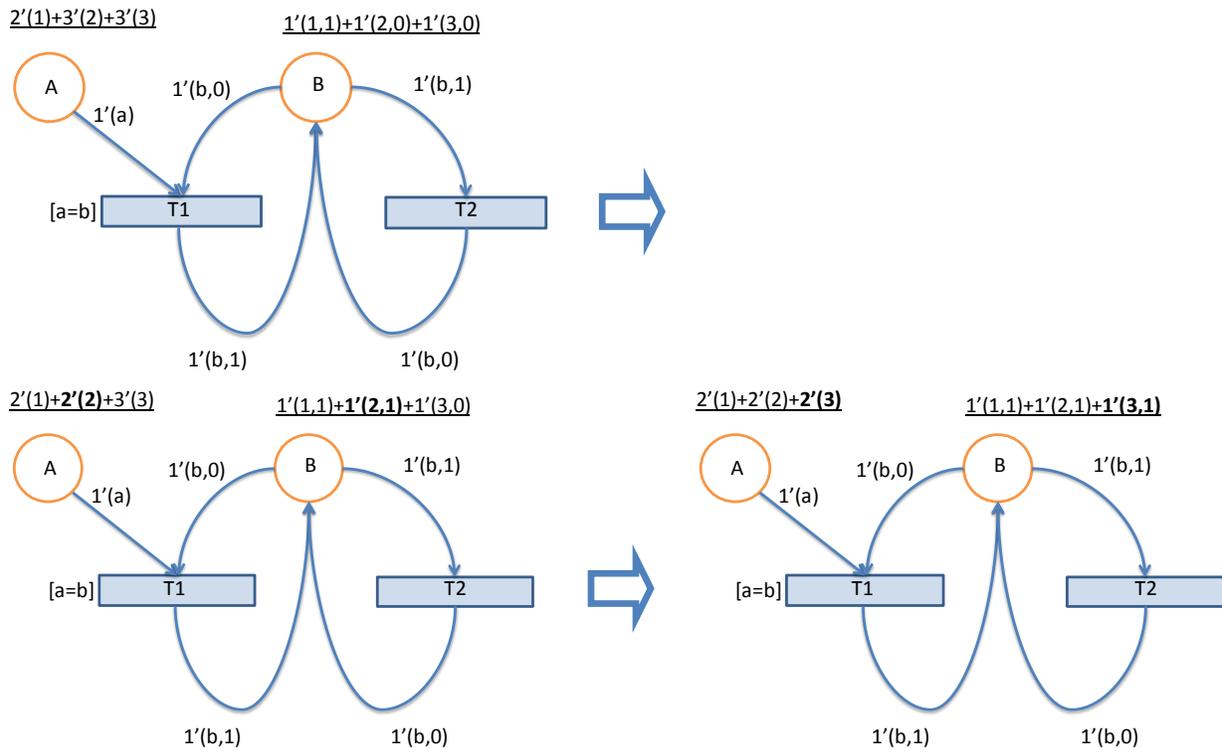


Figura 3.7 Dinámica en RPC.

$$M1=[2'(1)+3'(2)+3'(3), 1'(1,1)+1'(2,0)+1'(3,0)]$$

$$M2=[2'(1)+2'(2)+3'(3), 1'(1,1)+1'(2,1)+1'(3,0)]$$

$$M3=[2'(1)+2'(2)+2'(3), 1'(1,1)+1'(2,1)+1'(3,1)]$$

El estado global del sistema está dado por la simultaneidad de las acciones que han sido representadas secuencialmente. En este caso se ha representado únicamente el disparo de T1; sin embargo, a partir de M2, la transición 2 puede ser también disparada, representado que la estación de trabajo 2 ha sido desocupada: $M4=[2'(1)+2'(2)+3'(3), 1'(1,0)+1'(2,1)+1'(3,0)]$.

3.8 Redes de Petri Coloreadas Temporales

Las Redes de Petri pueden ser extendidas incorporando el concepto de tiempo a los modelos, éste concepto está basado en la introducción del reloj global. El valor del reloj representa el modelo de tiempo y cada marca tiene asociado un valor de tiempo, también llamado sello de tiempo. El sello de tiempo describe el modelo de tiempo más temprano en el cual una marca puede ser usada. Dicho de otra forma, el sello de tiempo de las marcas que serán eliminadas debe ser menor o igual que el modelo de tiempo actual.

Una vez que se han disparado todas las transiciones posibles para el actual modelo de tiempo, el reloj global se incrementa al modelo de tiempo más cercano en el cual al menos una transición

esté habilitada. Para modelar que una actividad correspondiente a una transición habilitada tomar unidades de tiempo, a las marcas añadidas a los nodos lugar de salida de la transición se le asocia un sello de tiempo el cual es r unidades de tiempo mayor que el modelo de tiempo en el cual se disparó la transición. Las marcas entonces no estarán disponibles durante r unidades de tiempo y no serán eliminadas por el disparo de transiciones antes de que el modelo de tiempo haya sido incrementado en al menos r unidades de tiempo [Jensen 1997; Kristensen y Christensen 2004].

En las Redes de Petri Coloreadas Temporales, se incorporaran elementos de tiempo al sistema:

- Reloj Global. Representa el tiempo antes de la ocurrencia de un evento o disparo de una transición.
- Valor de Tiempo. Elemento asociado a cada marca en un marcado, que describe el menor valor de tiempo en el cual una marca puede ser usada para habilitar una transición.
- Tiempo de retardo. Indica el tiempo que una transición consume de una actividad modelada.
- Tiempo de llegada. Es el tiempo en el cual el disparo de una transición cambia el estado del sistema a un nuevo estado.

En las redes temporales para que una transición pueda ser disparada, además de cumplir con las condiciones de arco, es necesario que todos los tiempos (sellos de tiempo) asociados a las marcas que habilitan la transición sean menores o iguales que el valor actual del reloj. En caso contrario el reloj global avanzará hasta el menor valor de sello de tiempo de las marcas para las cuales la transición está habilitada.

3.9 Análisis

El objetivo principal del análisis es determinar propiedades del modelo ligadas a su comportamiento dinámico:

- Determinación de bloqueos del sistema.
- Encontrar los posibles caminos para alcanzar un estado final partiendo de un estado inicial y obtener un coste de cada uno de los caminos.
- Obtener el conjunto de estados posibles a los que se puede llegar a partir del estado inicial.

Árbol de Alcance

Dada una RdP $N=(P,T,A,W,M_0)$ se desea determinar si existe una secuencia σ de transiciones que permitan alcanzar un estado M a partir de un estado inicial M_0 : $M_0 \xrightarrow{\sigma} M$.

El conjunto de estados M a los que se puede acceder a partir de un estado inicial M_0 , se determina con la función de alcanzabilidad (Reachability), $R(M_0)$. Para determinar este conjunto de estados se construye el árbol de alcance, el cual almacena en cada nodo del árbol un posible estado del

sistema, y los arcos que conectan un nodo con sus hijos indican la transición que debería dispararse para alcanzar el nuevo estado, las reglas de construcción del árbol son:

- La raíz del árbol es el estado inicial M_0 .
- Para cada nodo del árbol se generan tantos nodos hijos como transiciones hay activadas. Los nodos hijos representan el estado de la RPC una vez que se ha disparado cada una de las transiciones habilitadas. En los arcos que conectan los nodos padres con los nodos hijos se indica gráficamente la transición que se ha disparado y las marcas habilitadas.
- Cuando se genera un nodo hijo que ya existe en algún otro nivel del árbol diferente o igual al nivel actual se marca como nodo repetido (old) y no se generan nuevos nodos a partir de él, en el caso de Redes de Petri Temporales se considera que un nodo está repetido, cuando para dos marcados, son iguales tanto los colores como la información de tiempo asociada a cada marca (sello de marca) en ambos marcados.
- Un nodo del árbol que no tenga ninguna transición activada se marca como nodo hoja (dead end).
- Un nodo que no esté marcado como nodo repetido (old) ni como nodo hoja (dead end) se marca como nodo nuevo (new) para indicar que a partir de él es posible generar nuevos nodos hijos.

El árbol de alcance es una variante del árbol de cobertura que se emplea para analizar las Redes de Petri Coloreadas cuando no son acotadas.

Uno de los aspectos más importantes en el análisis y diseño de sistemas concurrentes que comparten recursos es garantizar que el sistema no entre en una situación de bloqueo. Las Redes de Petri Coloreadas son una buena herramienta para describir las interacciones lógicas entre los subsistemas que integran un proceso, ya que ofrecen toda la información necesaria para poder determinar, si un sistema correctamente expresado en Redes de Petri, puede entrar o no en estado de bloqueo. Más concretamente aquellos nodos del árbol de cobertura que no tienen ninguna transición activada, (nodos hoja ó dead end) describen aquellos estados del sistema, en los que una vez alcanzados, el sistema queda se detiene. Normalmente estos nodos indican que el sistema entra en una situación de bloqueo. El estado representado en el nodo hoja puede ser una solución factible o no factible.

Si se determina que el sistema se puede bloquear, a menudo es necesario incluir nuevas restricciones o eliminar restricciones existentes para evitar el bloqueo.

Al realizarse un análisis del sistema a través del árbol de cobertura, es posible conocer todos los estados posibles que el sistema puede alcanzar durante la evolución de mismo, no obstante, el tiempo computacional requerido para abrir el árbol puede ser elevado debido a todos los estados que se generan a partir del estado inicial.

Capítulo 4

MODELADO DE SISTEMAS LOGÍSTICOS EN REDES DE PETRI

La planificación de la producción como estrategia de control, consiste en un conjunto de planes sistemáticos y acciones encaminadas a administrar la producción, la cual considera entre otros factores, la configuración para que diferentes unidades a ser procesadas ingresen a estaciones de trabajo, es decir la secuencia en una producción en línea.

La secuencia en una línea de producción multiproducto determina en gran manera si el sistema es eficiente o no lo es. Así encontrar la configuración exacta en una secuencia con el fin de producir diferentes tipos de productos ha sido uno de los principales problemas en la industria manufacturera.

En este capítulo se presenta un mecanismo el cual accede al conocimiento para corregir la configuración de la secuencia en una línea de producción a través del intercambio de elementos en la línea misma.

4.1 Descripción del Problema de Secuencia

El problema de secuenciación se conceptualiza en este trabajo, como la búsqueda de una configuración a lo largo de una línea de producción multiproducto, de tal forma que al dirigirse los elementos de ensamble a las estaciones de trabajo especializadas, la distribución de piezas en cada línea sea la apropiada, lo que se traduce a que estas estaciones siempre estén activas y el stock de cada estación de trabajo sea diferente de cero.

Dada la importancia del problema en la industria de automóviles, se utiliza este sector como ejemplo y desarrollo del algoritmo de intercambio propuesto, la descripción simplificada del problema se presenta a continuación:

En una planta automotriz los vehículos son procesados en dos células de trabajo. Al inicio de las actividades, vehículos de diferentes tipos esperan a iniciar el primer proceso, este proceso es común para todos los vehículos en espera, los automóviles entran a este a la célula de trabajo con una frecuencia indistinta según su tipo, i.e. bajo una secuencia aleatoria; al salir de la primer estación de trabajo, cada vehículo es dirigido hacia la segunda célula según su tipo, si la célula está ocupada, los vehículos formarán parte del stock y esperaran hasta que la célula se encuentre desocupada, figura 4.1.

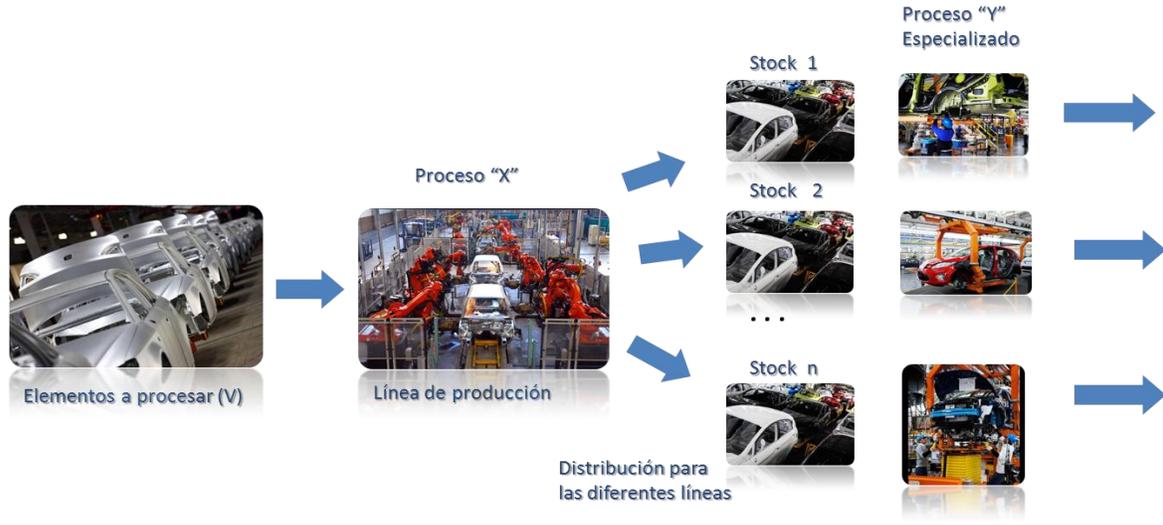


Figura 4.1 Esquema general.

Para que el proceso de producción sea eficiente, es necesario que todas las estaciones de trabajo se encuentren con vehículos en proceso, ya que una estación inactiva representa recursos desaprovechados. Ello conlleva la necesidad de elegir una secuencia inicial de vehículos que permita que al desocuparse la estación de trabajo exista al menos un vehículo en espera para ser procesado. Así la configuración inicial en la que los diferentes tipos de vehículos que han iniciado el primer proceso, será reflejada en los stocks de las siguientes estaciones de trabajo.

El presente modelo busca establecer un procedimiento basada en el intercambio de vehículos de tal forma que todas las líneas se encuentren produciendo, a menos que se hayan agotado las piezas para su producción, es decir que el stock inicial se encuentre vacío.

De acuerdo a la experiencia de la planta, cada día se conoce el número de vehículos a procesar, con todo, es posible que existan unidades que se hayan quedado a la mitad del proceso en el día anterior, estas unidades son procesadas al mismo tiempo en que las unidades nuevas entran a la primera célula.

4.2 Comportamiento de la Secuencia

En el problema de secuencia, las decisiones tomadas para establecer una configuración de entrada de las diferentes piezas en el sistema de producción pueden aparecer horas más tarde, las consecuencias se traducen en situaciones indeseables como la ruptura de stock o la inactividad de las máquinas. Éstos escenarios no productivos pueden ser identificados fácilmente con Redes de Petri Coloreadas.

Supongamos un sistema productivo donde en el stock inicial A, se tienen diferentes tipos de piezas, las cuales serán procesadas en una estación de trabajo B, frente a este sistema se debe tomar la decisión de cual tipo de pieza debe ser priorizada; posteriormente éstas piezas llegan al stock C, y serán dirigidas a los diferentes stocks por línea Tipo1, Tipo2, Tipo3, Tipo4, para que entren a las

estaciones de trabajo. En forma general se representa el sistema con la siguiente Red de Petri Coloreadas, figura 4.2.

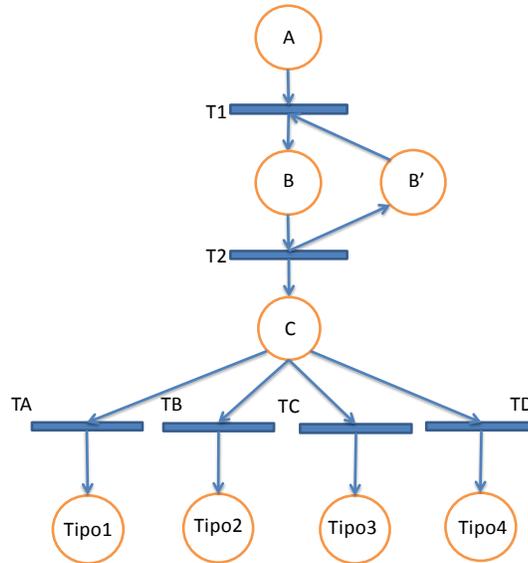


Figura 4.2 Red de Petri Coloreada de un sistema de producción.

Tabla de Especificaciones:

Nodo Lugar	Significado
A	Piezas en el Stock inicial A
B	Estación de Trabajo
B'	Transportador de Piezas
C	Stock de piezas procesadas

Nodo Lugar	Significado
Tipo1	Stock de la línea Tipo1
Tipo2	Stock de la línea Tipo2
Tipo3	Stock de la línea Tipo3
Tipo4	Stock de la línea Tipo4

Nodo Transición	Significado
T1	Una pieza del tipo i es transportada a la estación de trabajo i=1,2,3,4
T2	Una pieza procesada es dirigida al stock C
TA	Una pieza del tipo1 es dirigida al Stock Tipo1
TB	Una pieza del tipo2 es dirigida al Stock Tipo2
TC	Una pieza del tipo3 es dirigida al Stock Tipo3
TD	Una pieza del tipo4 es dirigida al Stock Tipo4

El método de evaluación para disparar una transición en RPC, con el objetivo de abrir el árbol de alcance, consiste en disparar la transición por cada posible combinación de marcas en los lugares de entrada. En caso de que esta combinación satisfaga todas las expresiones de arco y guardas,

la transición puede ser disparada y un nuevo nodo hijo (estado del sistema) es agregado al árbol de alcance.

El árbol de cobertura del sistema descrito cuando el stock inicial es de 40 piezas (10 de cada tipo) y se tienen 2 piezas en cada stock de Tipo $i=1,2,3,4$, es de tamaño factorial debido a las posibles combinaciones de disparar la transición T1 junto con las Transiciones TA, TB, TC y TD. Al abrir el árbol (figura 4.3), se encuentran los nodos hoja que surgen cuando no existe ninguna transición que pueda ser disparada, representando un camino no factible, ya que el objetivo del sistema es que todas las piezas sean procesadas sin que haya una ruptura de stock.

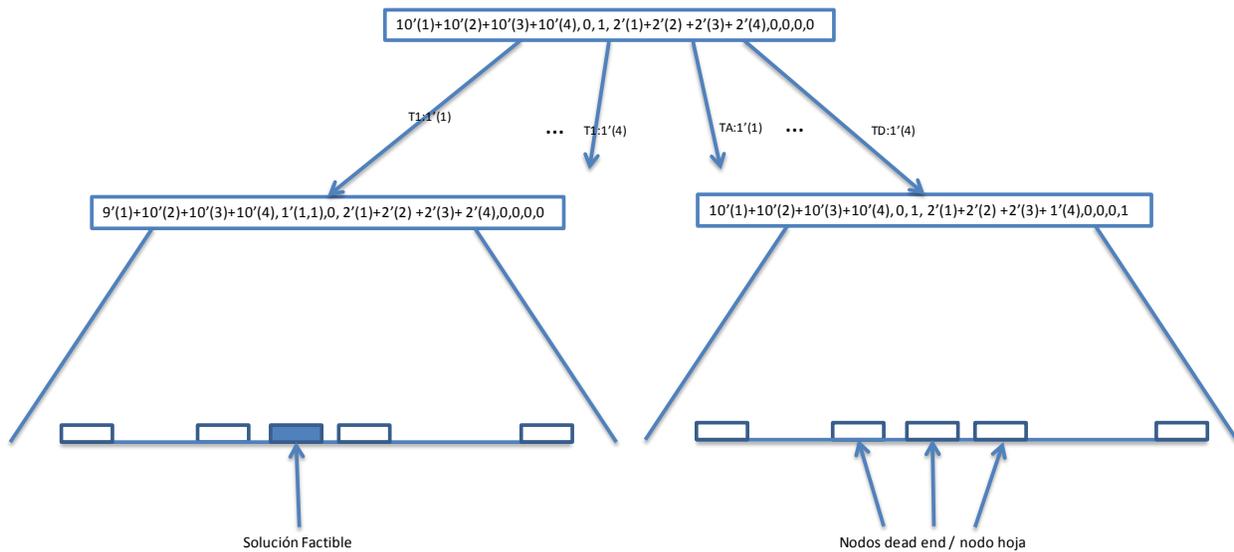


Figura 4.3 Representación del árbol de cobertura.

La principal desventaja de las RPC como formalismo para determinar las posibles configuraciones se encuentra en el tamaño del árbol de cobertura producido por un sistema complejo, como los sistemas logísticos. El objetivo de la programación en los sistemas logísticos, consiste en encontrar la secuencia de operaciones que permite la conducción del sistema desde su estado inicial hasta su estado final. No siendo posible construir árbol de cobertura completo, es necesario mecanismos para encontrar soluciones factibles.

4.3 Sistema de reparación

Para determinar si un nodo hoja en el árbol de cobertura es una solución factible, es necesario revisar todos los estados que produce la secuencia para determinar si en el nodo C, continuando con el sistema arriba descrito, existe al menos una marca que active las transiciones TA, TB, TC, TD. No obstante, no es sencillo determinar la secuencia se garantice que en los stocks 1 a 4, se encuentren siempre piezas para los siguientes procesos de acuerdo a su línea de producción o tipo. Para esta representación, el árbol de cobertura es muy grande partiendo y explorar todo, requiere de un tiempo computacional elevado, debido a todas las posibles combinaciones para elegir la secuencia.

El uso de guardas, ayuda a reducir el tamaño del árbol de cobertura. Cada vez que un nuevo estado es alcanzado, con el objetivo de continuar con la evaluación de sus estados hijos, todas las precondiciones asociadas a todos los eventos son evaluadas y en caso de encontrarse con un nodo hoja se realiza un “retroceso” al estado previo, y se trata de disparar otro evento con una configuración diferente de marcas.

Agregando a las transiciones un guarda que revise el número de marcas en el nodo lugar C, es factible reducir el espacio de búsqueda. Así cuando un estado “no deseado” es generado, sus hijos no son generados ya que la transición no está activada, acotando de esta forma el árbol de cobertura.

Un nodo hoja que no es solución factible aparece como consecuencia de una mala decisión cuando se elige la combinación de marcas que activaron un evento disparado anterior. En un sistema de producción es sencillo que un estado hoja, que es alcanzado al elegir una pieza incorrecta (supongamos tipo1), pueda ser el mismo que uno alcanzado eligiendo el tipo de pieza correcta (tipo 2, por ejemplo).

Un algoritmo de intercambio/reparación puede ser aplicado de tal manera que un nodo hoja, que no sea solución factible, sea transformado en un nodo que puede ser alcanzado si la pieza correcta se hubiese elegido (tipo 2). Este intercambio de estado se consigue implementar si cada vez que se toma una decisión, toda la información de las marcas candidatas es almacenada como parte de los datos (atributos) de la marca elegida.

Esta idea se encuentra basada en el hecho de que una mala decisión no puede ser detectada hasta que una restricción es violada. Lo cual algunas veces ocurre después de un largo periodo de haber sido tomada la decisión.

4.4 Modelo Gráfico

Seguendo el con el sistema descrito, se representa a continuación la RPC básica del sistema en el software CPN Tools (figura 4.4).

Conjuntos de Color:

- Tipo = int with 1..5 timed
- T = int
- C = int with 1..50 timed
- Prod = product Tipo*C timed
- Maq = with ocupado|libre;
- M = int with 0..10;

Variables:

- var x,r:Tipo;
- var y,i,h:C;
- var m,n,p,q,w:C;

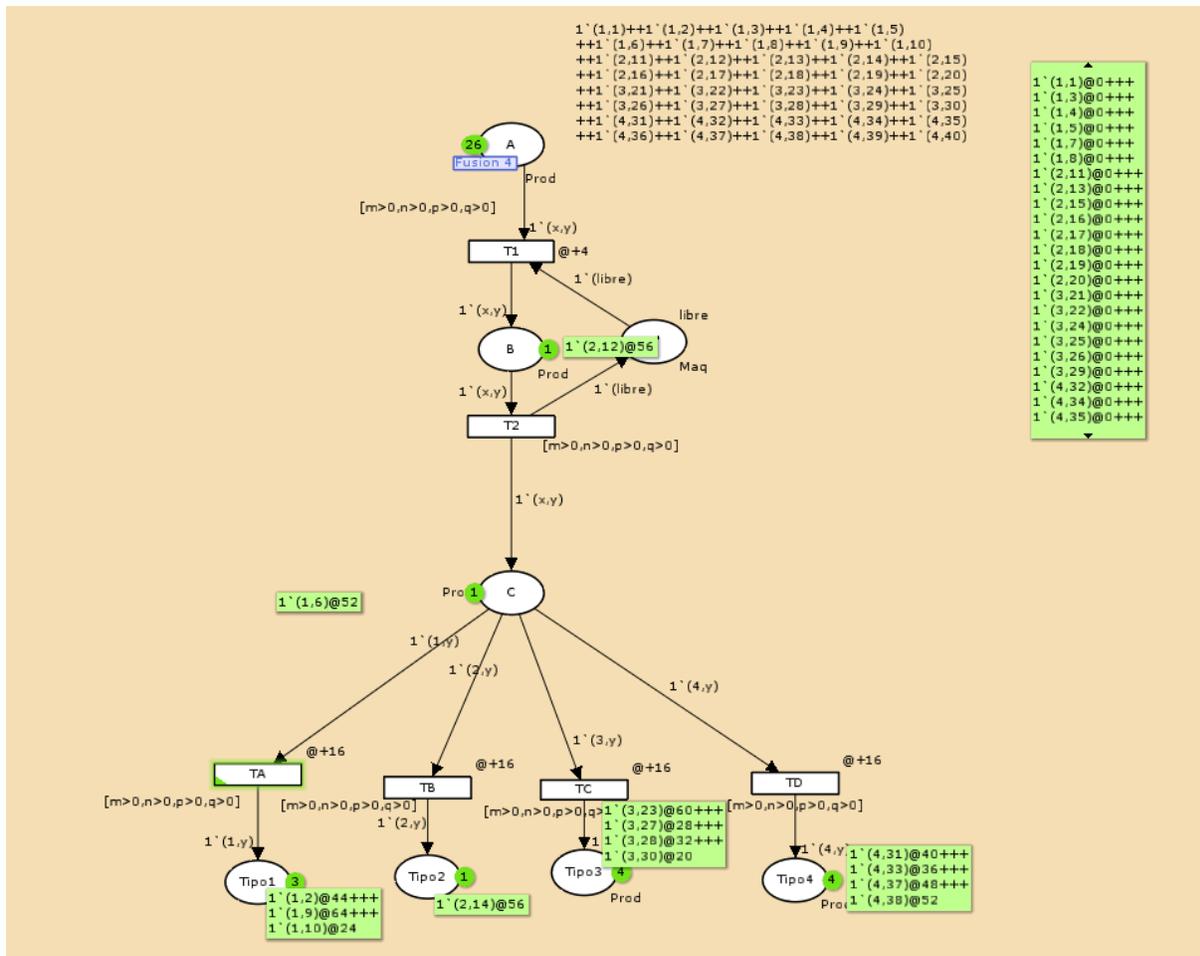


Figura 4.4 Red base del sistema.

4. Modelado de Sistemas Logísticos

Cada vez que lleguen las piezas al lugar C, activarán una transición (T14), la figura 4.5 se muestra ésta transición, la cual tiene por objetivo intercambiar las piezas cuando en el sistema se incline a un determinado tipo de pieza.

En el caso de llegar sólo un tipo de piezas, la transición efectuará un reemplazo de piezas, el color de la marca (token) que sea más numeroso se intercambiará por aquel color que haya caído a cero:

```
TransAct(m,n,p,q,h:C)=if (m>=n andalso m>=p andalso m>=q) then 1^(1,h)
else if (n>=m andalso n>=p andalso n>=q) then 1^(2,h)
else if (p>=m andalso p>=n andalso p>=q) then 1^(3,h)
else if (q>=m andalso q>=n andalso q>=p) then 1^(4,h)
else 1^(5,h)
```

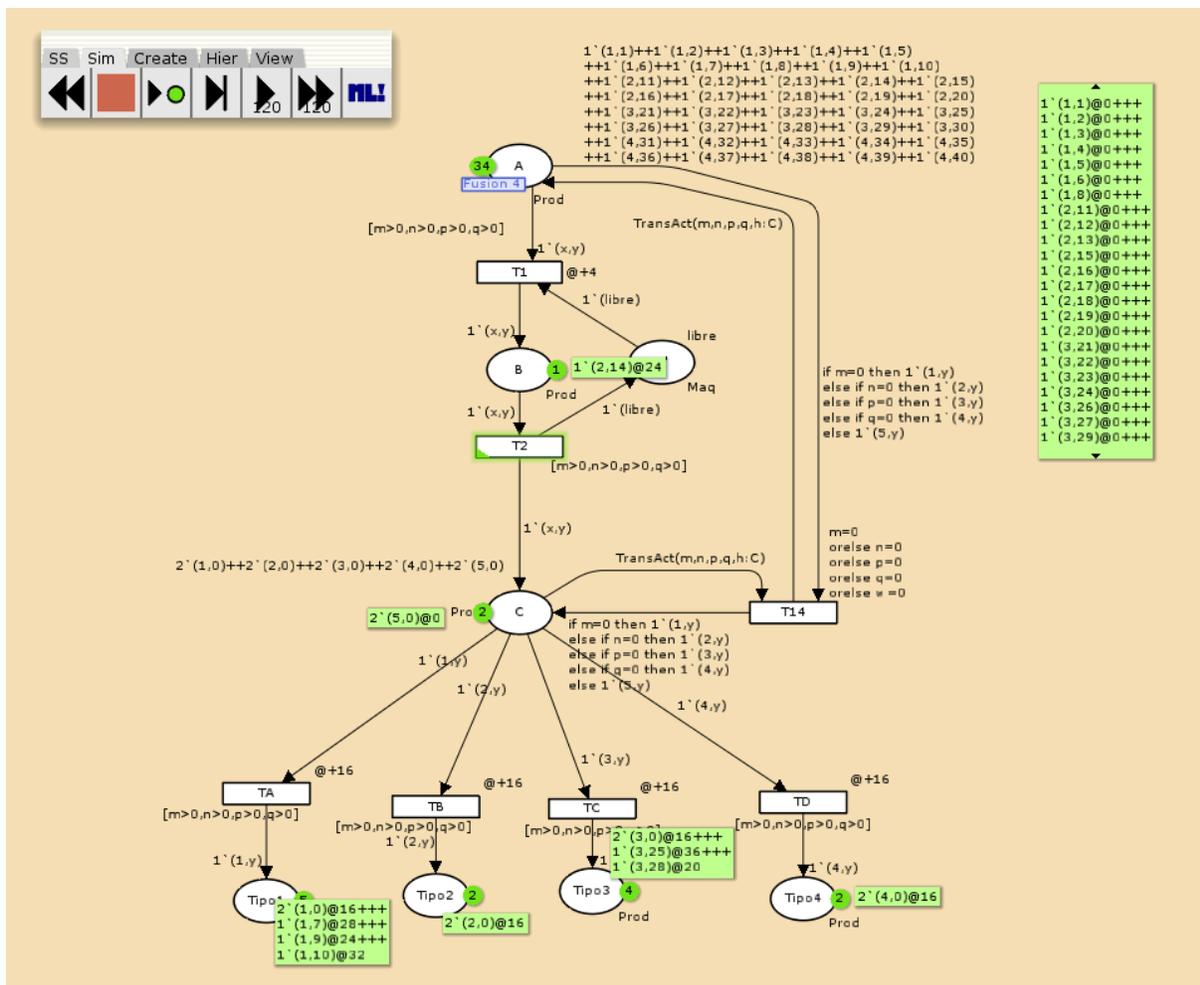


Figura 4.5 Red con transición de intercambio T14.

4. Modelado de Sistemas Logísticos

Se crea un nodo lugar G (figura 4.6) con un marcado inicial de dos marcas (tokens) para cada color que realizará las siguientes funciones:

- Contabiliza el número y tiempo de marcas (tokens) consumidos de cada color.
- Administra el intercambio de piezas.

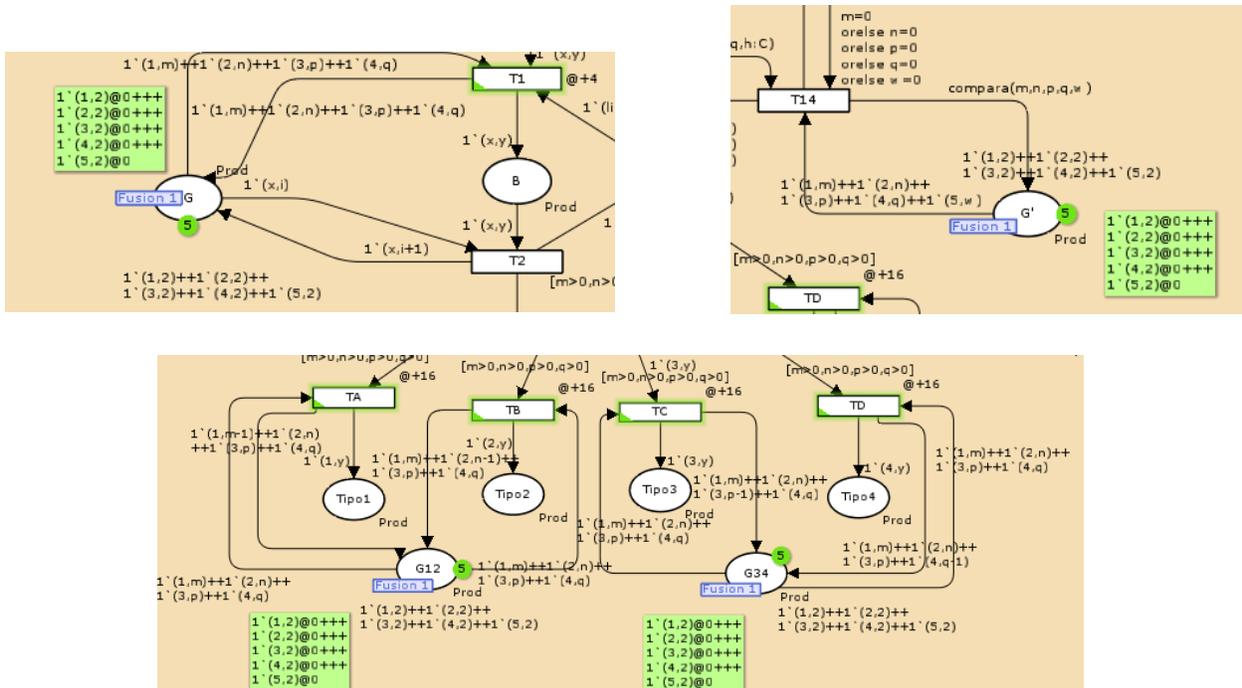


Figura 4.6 Inclusión del nodo administrador G.

4. Modelado de Sistemas Logísticos

El nodo C albergará inicialmente dos marcas (tokens) de cada color, controlará la salida de piezas de cada tipo, si cualquiera de las piezas cae a cero, se activará la transición de intercambio (T14), a continuación la figura 4.7 muestra la red con la transición de intercambio y el nodo administrador G.

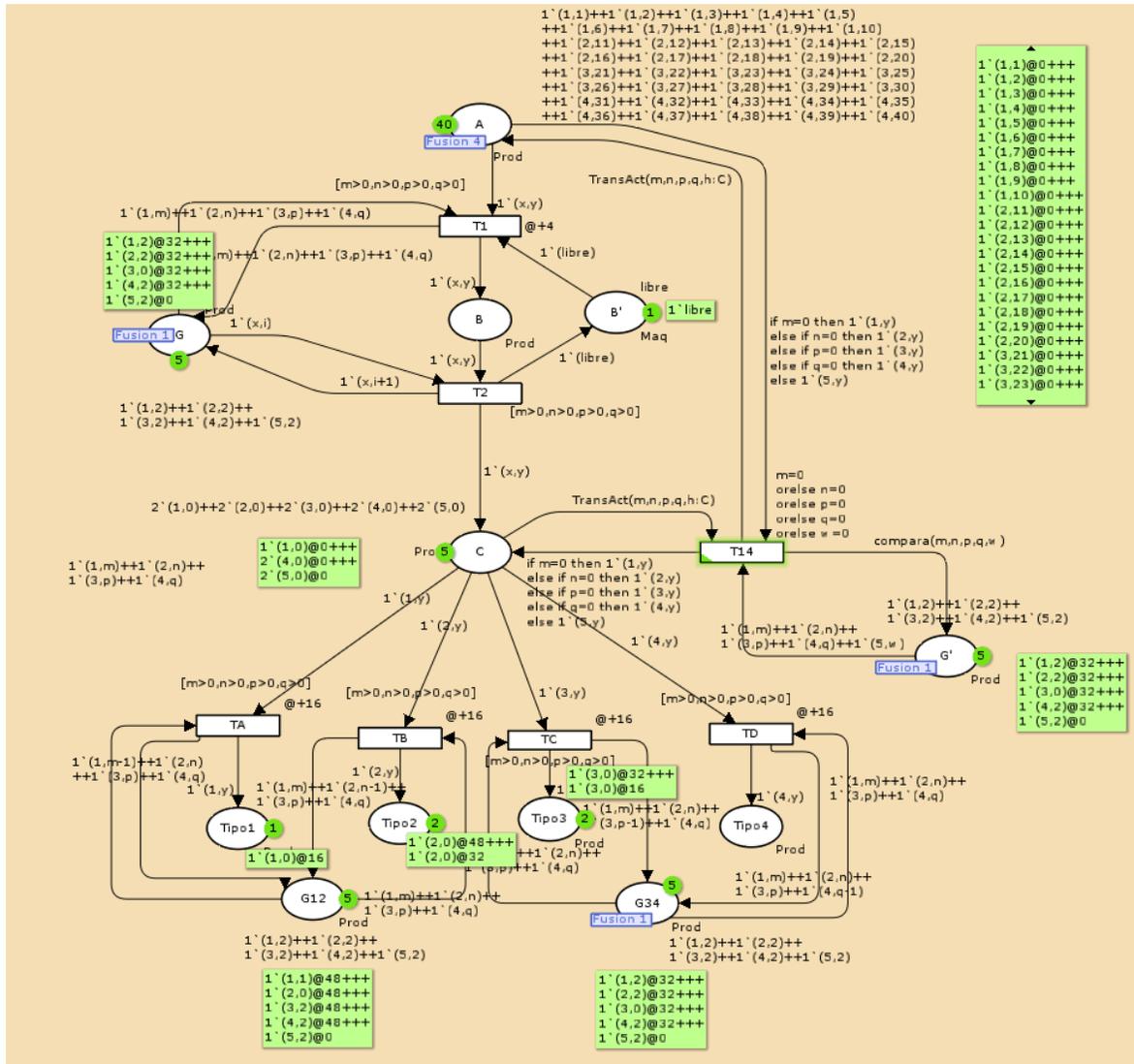


Figura 4.7 Red con transición de intercambio y nodo administrador G.

4. Modelado de Sistemas Logísticos

Sin embargo, cuando existe una sola marca (token) en cada color, el sistema intercambia el color que ha caído a cero con otro color, y al hacerlo otro más cae a cero y así sucesivamente, como se muestra en la siguiente figura 4.8:

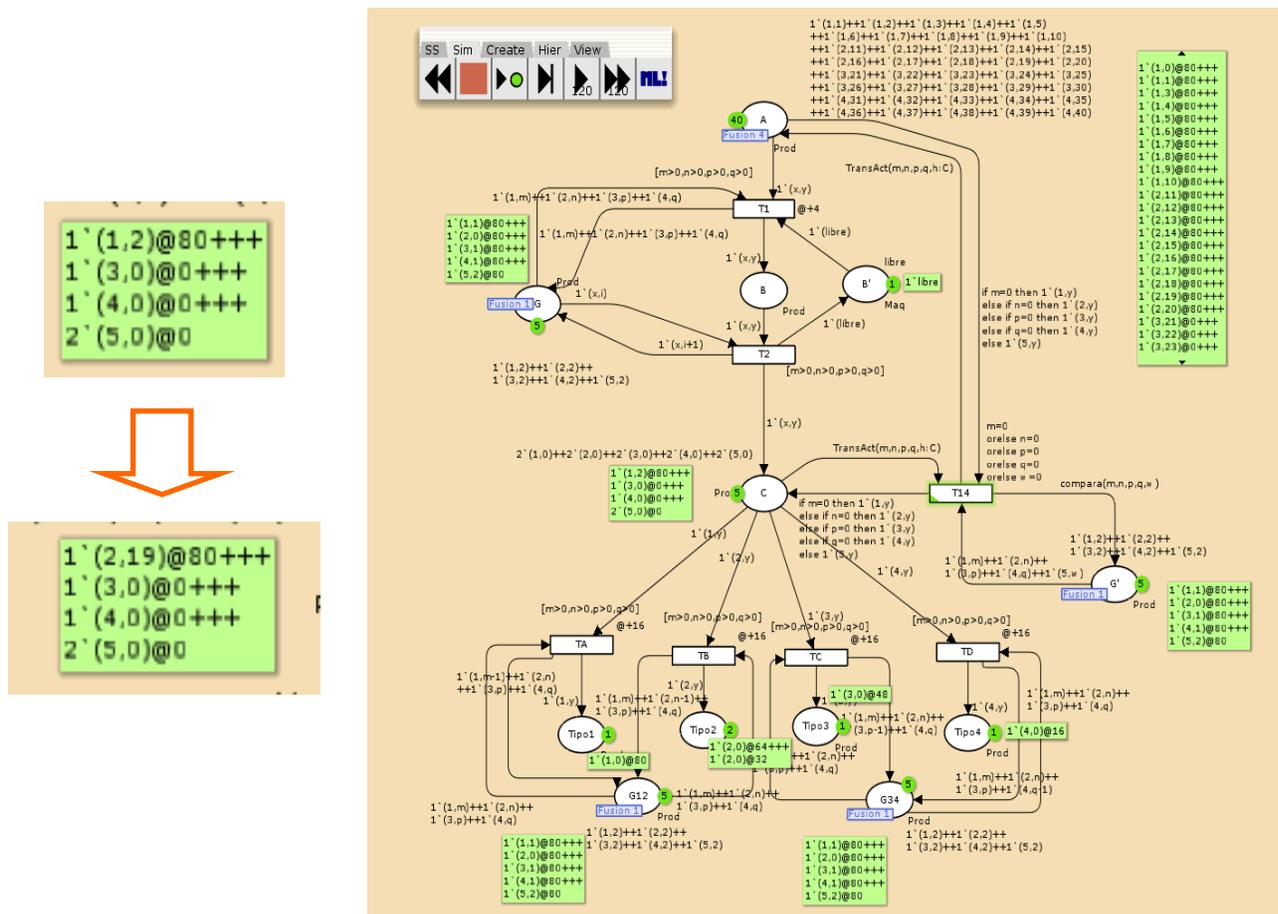


Figura 4.8 Representación de stock vacío por intercambio.

Se anexa a la red un nodo lugar reactivo **dd**, (figura 4.9) el cual tiene por objetivo dirigir del disparo de transiciones y resolver el conflicto cuando todos los colores tienen una marca (token) y algún color cae a cero.

El nuevo nodo activará las transiciones TA, TB, TC, TD correspondientes a la clasificación de las piezas, cuando se hayan acumulado 4 piezas de cada color, heredadas de T1, obligando de esta forma a que primero sea disparada la transición T1, relacionada con el proceso común y posteriormente las destinadas a la clasificación.

Se irán acumulando las piezas en el nodo C (figura 4.10), hasta que puedan ser disparadas TA, TB, TC, TD, cuando ninguna de estas transiciones pueda ser disparada debido a que alguno de los colores ha caído a cero en el nodo C, se activará T14, que intercambiará el tipo de pieza más

4. Modelado de Sistemas Logísticos

numeroso por el tipo de pieza con cardinalidad cero, de esta forma se tienen más de una marca (token) de cada color y se logrará el intercambio con éxito.

TA, TB, TC, TD no pueden ser disparadas ya que la cardinalidad de los colores 1 y 2 es cero en el nodo "reactivo". El color "2" ha caído a cero en C, condición necesaria para activarse la transición de intercambio.

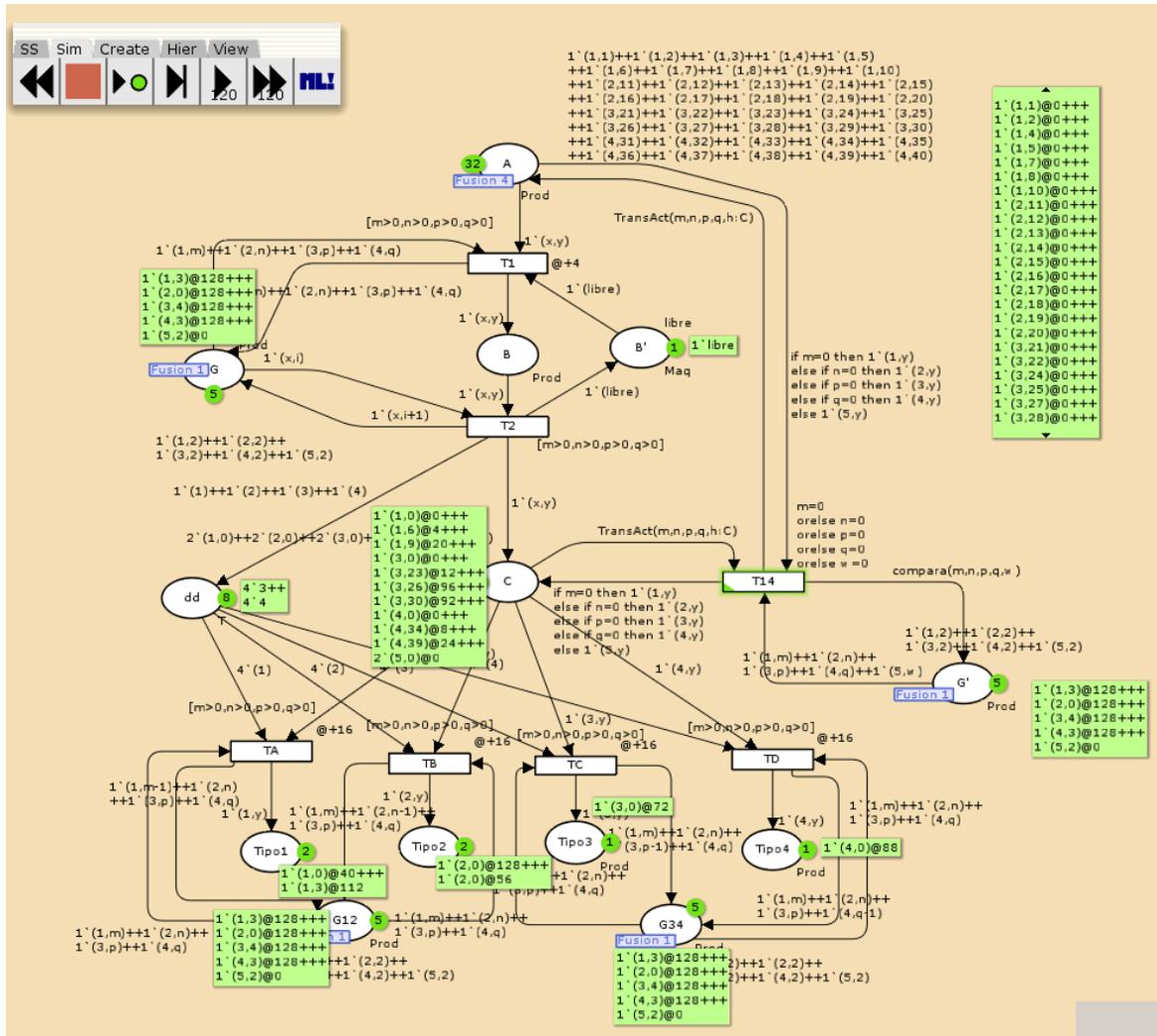


Figura 4.9 Red con nodo reactivo dd.

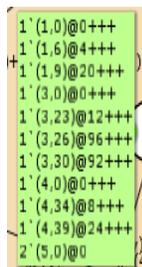


Figura 4.10 Acumulación de piezas en C.

Capítulo 5

PROBLEMA DE SECUENCIA. ALGORITMO DE REPARACIÓN

5.1 Definición Formal

Se considera un conjunto I de máquinas y un conjunto J de trabajos. Se atribuye que cada trabajo j en J consiste en un conjunto ordenado de operaciones de K_j cada una de las cuales debe ser procesada ininterrumpidamente sobre una máquina determinada durante un tiempo conocido t_k . El orden en el conjunto de operaciones de cada trabajo, implica que cada operación no puede comenzar a ser procesada hasta que todas las operaciones precedentes terminen de ser procesadas. Cada máquina sólo puede realizar una única operación a la vez y ésta debe realizarla sin interrupción [Salazar González, 2001].

Los problemas de secuencia se originan en los más diversos sectores, en el caso particular de este trabajo, se adapta la definición anterior considerando que el número de automóviles a producir de cada tipo j en J consiste en un conjunto ordenado de elementos de V cada uno de los cuales debe ser procesado sobre una máquina determinada durante un tiempo conocido $TP(x)$. El orden en el conjunto de elementos, implica que cada elemento no puede comenzar a ser procesado hasta que el elemento precedente termine de ser procesado en la máquina correspondiente. Cada máquina sólo puede realizar una única operación a la vez y ésta debe realizarla sin interrupción.

De esta forma se trata de secuenciar el conjunto de elementos que pasaran a lo largo de la línea de ensamble de manera que la finalización del proceso de los diferentes tipos de automóviles sea lo menor posible, evitando que células de trabajo para todas las líneas se encuentren inactivas. Se muestra un esquema general de la producción en línea con células especializadas en la siguiente figura 5.1:

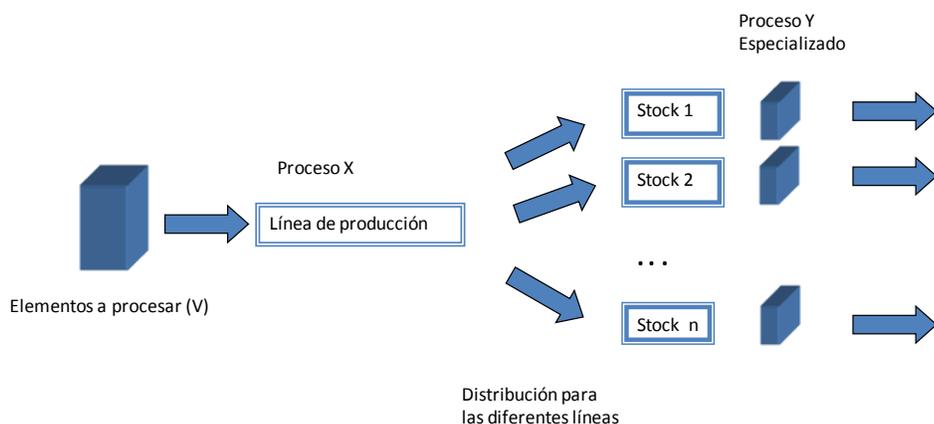


Figura 5.1 Esquema del problema de secuenciación.

5.2 Algoritmo

Se define el conjunto de vehículos como $V = \{x_{i,j} \mid 0 < i \leq m_j \mid 0 < j \leq n\}$, donde m_j es el número de vehículos del tipo j y n representa los diferentes tipos de automóviles. Se identifica el número de vehículos de cada tipo: $V_j \subset V$ conjunto de vehículos del tipo j en V , tal que $|V_j| = m_j$.

En el conjunto V , cada x_{ij} representa un vector con la información asociada al vehículo de tipo j con identificador i , por simplicidad V puede ser escrita como matriz tridimensional².

Definimos el arreglo matricial para $V=[a_{i,j}]_{M \times n}$:

$$a_{i,j} = \begin{cases} x_{i,j} & i = 1, \dots, m_j, |V_j| \leq M \\ 0 & i = m_j, \dots, M, |V_j| > M \end{cases}$$

Donde la cardinalidad para cada tipo es dada por $|V_j| = m_j$ y $M = \text{Max}\{|V_j| = m_j; j = 1, \dots, n\}$.

Entonces:

$$V = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,j} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,j} & \dots & x_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{i,1} & x_{m_2,2} & \dots & x_{i,j} & \dots & x_{i,n} \\ \dots & 0 & \dots & x_{m_j,j} & \dots & \dots \\ x_{M,1} & 0 & \dots & 0 & \dots & x_{M,n} \end{bmatrix}$$

Así, el número total de unidades que definen la carga de vehículos a secuenciar es $|V| = \sum_j |V_j| = z'$.

Al inicio de la jornada de producción se desconoce cuál son las secuencias óptimas de los vehículos, para interpretar lo anterior definimos $f: N \rightarrow N$ una función que asigna a cada elemento de V , un número aleatorio k diferente (pointwise function evaluation).

Dado $V=[a_{i,j}]_{M \times n}$, $f(x_{i,j}) := \begin{cases} k_{i,j} & x_{i,j} \neq 0 \\ 0 & x_{i,j} = 0 \end{cases}$ donde $k_{i,j} \in \{1, \dots, z'\}$ $k_{i,j} \neq k_{i',j'}$ $i \neq i', j \neq j'$

² Cada entrada de la matriz, representa un vector con la información (lugar en las diferentes secuencias, tiempos de inicio y fin de los procesos) correspondiente al vehículo tipo j con identificador i .

5. Problema de Secuenciación

$$V = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1,j} & \dots & x_{1,n} \\ x_{21} & x_{22} & \dots & x_{2,j} & \dots & x_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{i,1} & 0 & \dots & x_{i,j} & \dots & x_{i,n} \\ \dots & 0 & \dots & x_{m,j} & \dots & \dots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix} \Rightarrow f(V) = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1,j} & \dots & k_{1,n} \\ k_{21} & k_{22} & \dots & k_{2,j} & \dots & k_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ k_{i,1} & 0 & \dots & k_{i,j} & \dots & k_{i,n} \\ \dots & 0 & \dots & k_{m,j} & \dots & \dots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix} = L$$

Sea $h_s : N \rightarrow N$, que proporciona el elemento de la matriz de menor valor en L , ($s=1, \dots, z'$),

$$h_s \doteq \begin{cases} h_1 = k_{i,j} \ni k_{i,j} = \min\{k_{i,j} \neq 0 \in L\} & s = 1 \\ h_s = k_{i,j} \ni k_{i,j} = \min\{k_{i,j} \neq 0, \in L - [h_p, p = 1..s - 1]\} & s > 1 \end{cases}$$

Así x_{ij} , tendrá como primera entrada del vector información a $h_s(k_{ij})$ que representa la posición de entrada al proceso de producción (Secuencia 1).

En el problema de secuenciación de automóviles se establecen estaciones de trabajo en las cuales son procesadas cada una de las unidades, los vehículos pasan por las estaciones: Body Shop, Paint Shop y Assembly [Solnon, Cung, V-D., Nguyen, Artigues, 2008], figura 5.2.

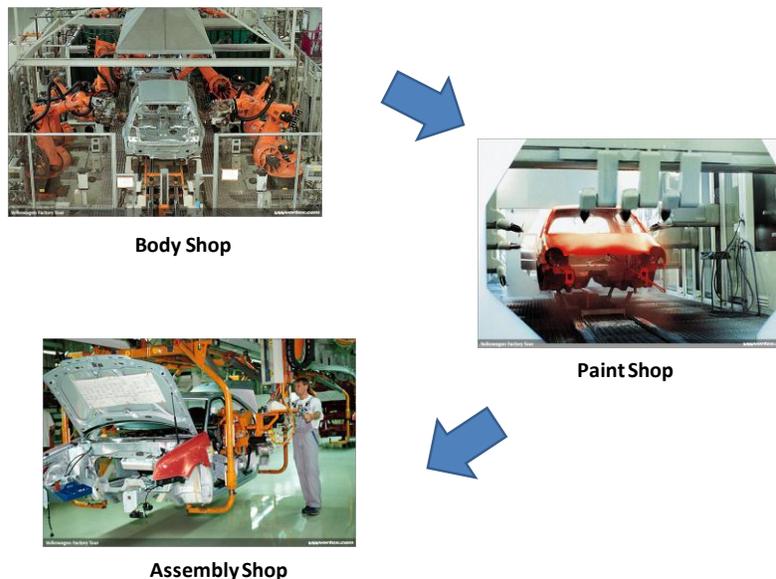


Figura 5.2 Estaciones de trabajo.

Sin pérdida de generalidad se tomarán dos procesos: el proceso “X”, que representa una célula de trabajo en la cual deberán ser procesados todos los vehículos y proceso “Y_j”, que se refiere a la línea de ensamble/proceso especializado por tipo de vehículo.

Cada vehículo requiere de “zx” unidades de tiempo para finalizar con el proceso X, en este proceso el tiempo requerido en la estación de trabajo es el mismo para todos los tipos de vehículos; el tiempo en la estación Y por tipo de vehículo está dado por: “zy_i”

Para llevar un registro de los tiempos en que cada unidad es procesada, a cada entrada de la matriz V , se le asigna la información de entrada y salida del proceso X , de acuerdo al lugar establecido por la secuencia derivada de las funciones f y h .

Entonces para cada unidad de la secuencia, se tiene $x_{i,j} = (h_s, ipx_s, fpx_s)$.

Tiempo inicial del proceso X

$$ipx_s(x_{i,j}) := \begin{cases} 1 & s=1 \quad [f(x_{i,j}) = k_{i,j} \ni h_1 = k_{i,j}] \\ fpx_{s-1}(x_{i,j}) + 1 & s > 1 \quad [f(x_{i,j}) = k_{i,j} \ni h_s = k_{i,j} \quad y \quad s \in \{2, \text{Max}(h_t) = z'\}] \end{cases}$$

Tiempo final del proceso X

$$fpx_s(x_{i,j}) := ipx_s + z_s, \quad s = 1 \dots z'$$

Definimos el conjunto V_A de cardinalidad r ($|V_A| = r$), que representa a las unidades que no se terminaron de procesar el día anterior, es decir, que sólo pasaron por la estación del proceso común, estos vehículos deben incorporarse a la secuencia que se ha registrado para el conjunto V , de tal forma que:

$$V \cup V_A = V_T \quad \ni |V_T| = r + z' = z$$

Al igual que el conjunto V , este conjunto V_A alberga diferentes tipos de vehículos $x^a_{i,j}$; donde ma_j es el número máximo de cada tipo que se encuentran a la mitad del proceso:

$$V_A = \{x^a_{i,j} \mid M < i' \leq ma_j \quad 0 < j \leq n, \},$$

A estas unidades se les ha asignado una secuencia de acuerdo a su tipo el día anterior, mismo que se conservará para continuar con el proceso Y_j , esta secuencia es inicializada en uno. La información correspondiente al tiempo de proceso X para estas unidades, será igual a cero, ya que ha sido realizado éste proceso el día anterior:

$$x^a_{i,j} = (h_s, ipx = 0, fpx = 0), \quad \text{con } s' = 1, \dots, r, \quad y \quad h_{s'} : \text{Secuencia del día anterior}$$

El conjunto total tendrá una nueva secuencia basada en los vehículos anteriores:

$$x_{i,j} = (h_{2s}, ipx, fpx), \quad \text{donde } 2s = 1, \dots, r, \quad 1+r, \dots, z \quad y \quad x_{i,j} \in V \cup V_A = V_T$$

h_{2s} : Secuencia incorporando todos los automóviles.

Cuando se han establecido los tiempos para cada unidad x_{ij} , de entrada y salida del proceso X en base a la secuencia establecida de forma aleatoria, se incorpora la información de entrada y salida para cada unidad del proceso Y , para ello es necesario la clasificación por tipo, ya que cada proceso j tiene diferentes tiempos de proceso.

Para la clasificación por tipo de pieza dentro de la secuencia s_2 se crean subconjuntos:

$$Tp(j) = \{h_{2s} \ni h_{2s} := k_{i,j} \quad i = 1, \dots, m_j\}$$

En cada subconjunto (tipo de auto $j=1, \dots, n$), se establece el orden de la secuencia de acuerdo al valor mínimo de h_{2s}

$$h'_{s(j)} \doteq \begin{cases} \min\{Tp(j)\} & s = 1 \\ \min\{Tp(j) - h'_{s-1(j)}\} & s > 1 \end{cases}$$

$h'_{s(j)}$: Secuencia en la línea de producción j

Sean $zy(j)$, $1 \leq j \leq n$, el tiempo en que tarda en procesarse una unidad del tipo j en el proceso y , para cada subconjunto.

Tiempo inicial del proceso Y en j ($x_{i,j} \ni f(x_{i,j}) = k_{i,j}; k_{i,j} := h'_{s(j)}$),

$$ipy_{s(j)}(x_{i,j}) := \begin{cases} 1 & s = 1 \\ fpy_{s-1(j)} + 1 & s > 1 \\ \max\{fpy_{s-1(j)}, fpx_s\} & s > 1 \end{cases} \quad \begin{matrix} fpx_{s(j)} = 0 \\ fpx_{s(j)} \neq 0 \end{matrix}$$

Tiempo final del proceso Y del tipo de auto j ,

$$fpy_{s(j)} := ipy_{s(j)} + zy(j) \quad \Rightarrow \quad x_{i,j} = (h_{2s}, ipx, fpx, h'_{s(j)}, ipy_j, fpy_j)$$

El establecimiento del tiempo de entrada y salida de cada proceso para cada vehículo, lleva consigo el concepto de "cola", el objetivo en este modelo es la continuidad de las líneas de producción y en consecuencia la existencia piezas en espera, o al menos vehículos en proceso en todas las líneas.

Colas en t

De acuerdo al valor para cada una de las variables de tiempo (ipx , ifx , ipy , fpy) se establece el estatus del vehículo: en espera, en proceso, o procesado; de igual forma se puede determinar el estatus de la estación de trabajo: vehículos en proceso, línea vacía o stock vacío. En cada tiempo t se tiene para las diferentes líneas:

$$t = (x_{i,j}, h_{2s}, ipx, fpx, h'_{s(j)}, ipy_j, fpy_j), j=1 \dots n$$

Para conocer si el tipo de pieza $x_{i,j}$ hace cola en el tiempo t , en la línea j , sea $C_t(x_{i,j})$ variable binaria como sigue:

$$C_i(x_{i,j}) = \begin{cases} 1 & fpx(x_{i,j}) < t \leq fpy(x_{i,j'}) \quad \exists x_{i,j} := h_{s(j)}, x_{i,j'} := h_{s(j)-1} \\ 0 & eoc \end{cases}$$

La acumulación de esta variable, dará el tamaño de la cola en el tipo j: $C_i(j) = \sum_{i=1}^{m_j} C_i(x_{i,j})$.

Estatus de estación de trabajo

La evaluación de las estaciones de trabajo se realizan bajo dos consideraciones de tiempo: TG y C.

La variable TG establece el periodo de “gracia” en el cual se da tiempo a que comience a funcionar el sistema que esta en función del número de piezas a mitad del proceso (x_{ij}^a) en cada línea y el tiempo que tarden según su tipo $zy(j)$, de tal forma que a partir de la siguiente unidad de tiempo de TG se establece el estatus de la estación de trabajo.

$$TG = \text{Max}\{|V_A(j)| * zy(j), j = 1, \dots, n\}$$

Durante TG el sistema no reportará el estatus de línea vacía, será hasta la llegada del primer auto para todas las líneas, cuando iniciará la evaluación de colas para cada sistema.

Para $t > Tg$ se tiene:

Estación ocupada	$\exists x_{i,j} \ni ipy(x_{i,j}) \leq t \leq fpy(x_{i,j})$
Piezas en cola	$\exists x_{i,j} := h'_{s(j)}, x_{i',j''} := h'_{s(j)+1} \ni \begin{matrix} ipy(x_{i,j}) \leq t \leq fpy(x_{i,j}) \\ fpx(x_{i',j''}) < fpy(x_{i,j}) \end{matrix}$
Línea vacía	$\exists x_{i,j} := h'_{s(j)}, x_{i',j''} := h'_{s(j)+1} \ni \begin{matrix} fpy(x_{i,j}) < t \\ ipy(x_{i',j''}) > t \end{matrix}$
Stock Vacío	e.o.c.

Iniciado el proceso de producción existe la probabilidad de que alguna línea de ensamble tenga productividad cero (línea vacía), y por otra parte otras líneas este sobre cargadas (colas), lo que reflejaría que la subsecuencia elegida de forma aleatoria no ha sido lo suficientemente “buena” para la programación de las líneas de ensamble.

La elección “buena” de la secuencia, lleva consigo la utilización de métodos heurísticos para la búsqueda de la secuencia, se establece una metodología de intercambio y reparación, seleccionando las piezas que dada su secuencia ocasionan el desequilibrio del sistema.

Mediante un sistema de “reparación” se propone intercambiar unidades en la línea de espera más “tardada” por unidades cuya línea de espera se encuentra vacía.

Identificación de elementos a intercambiar

Se evalúa en cada tiempo t , el estatus de las estaciones, supongamos t^* , tal que la línea de producción ha caído a cero, es decir,

Para $t^* \neq TG_j, \forall j$:

$$\exists x_{i,j} := h'_{s(j)}, x_{i',j'} := h'_{s(j)+1} \ni \begin{matrix} fpy(x_{i,j}) < t \\ ipy(x_{i',j'}) > t \end{matrix}$$

Entonces $x_{i',j'}$ se identifica como la pieza causante de la caída a cero, para $j=1, \dots, n$ ya que dado su lugar en la secuencia $h_{s(j)+1}$, termina el proceso X e inicia el proceso Y , después de haberse terminado de procesar en Y la pieza anterior, sea $x_{i',j'} \doteq x_{i^*,j^*}$; en caso de caer más de una línea a cero, se toma de acuerdo al *orden numérico de las líneas*.

Por otro lado se evalúan las piezas en el tiempo t^* para las diferentes líneas identificando la cola más "tardada", la cual se determina a través del tamaño de la cola y el tiempo en que tarda el proceso en cada línea:

Dado $C_{t^*}(j) = \sum_{i=1}^{m_j} C_{t^*}(x_{i,j})$, definimos $C_{t^*}^+(j) = \max\{C_{t^*}(j), j=1, \dots, n\} * zy(j)$, como la línea más tardada en el tiempo t^* , es decir la línea que tardará más tiempo en consumirse debido a tiempo del proceso en referencia y al tamaño de la línea de espera.

Sea $j^c \ni C_{t^*}^+(j) = C_{t^*}(j^c) * zy(j^c)$ en el momento t^* , entonces se identifica la pieza que hace cola en la línea j^c para intercambiar su secuencia con x_{i^*,j^*} :

$$t^* = (x_{i,j^c}, h_{2s}, ipx, fpx, h'_{s(j^c)}, ipy, fpy), \Rightarrow ipy(x_{i,j^c}) \leq t^* \leq fpy(x_{i,j^c}),$$

Es decir la pieza x_{i,j^c} con la secuencia $h'_{s(j^c)}$ se encuentra en proceso, y siguiente pieza en la secuencia $x_{i',j'^c} := h'_{s(j^c)+1}$ se encuentra en cola, sea $x_{i',j'^c} \doteq x_{i^c,j^c}$

Procedimiento de Intercambio

Para $t^* \exists j^* := \text{caída a cero y } j^c := \text{cola más tardada}$,

$$t^* = (x_{i^*,j^*}, h_{2s}, [x_{i^*,j^*}], ipx, fpx, h'_{s(j^*)} [x_{i^*,j^*}], ipy, fpy)$$

$$t^* = (x_{i^c,j^c}, h_{2s}, [x_{i^c,j^c}], ipx, fpx, h'_{s(j^c)} [x_{i^c,j^c}], ipy, fpy)$$

Se intercambiará la secuencia del vehículo $x_{i^*,j^*} := h_{2s}$, con la secuencia de $x_{i^c,j^c} := h_{2s}$.

Sea $I_{ct^*} : h_{2s} \rightarrow h_{2s}$, la función de intercambio de la secuencia asignada a cada x_{ij} ,

$$I_{c^t} [h_{2s}(x_{ij})] = \begin{cases} h_{2s}(x_{i,j}) & i \neq i^c, i^* & j \neq j^c, j^* \\ h_{2s}(x_{i^*,j^*}) & i = i^c & j = j^c \\ h_{2s}(x_{i^c,j^c}) & i = i^* & j = j^* \end{cases} \quad \text{intercambio c, en el tiempo } t^*$$

Al ir avanzando el proceso, es necesario indicar cuales piezas han sido consumidas o están a punto de finalizar su proceso, en cada intercambio se evalúan la secuencia de las unidades, marcando aquellas que han salido del sistema:

$$sl_t[x_{ij}] = \begin{cases} M & ipy(x_{i,j}) > t \\ h_{2s}(x_{ij}) & e.o.c \end{cases}$$

Cada vez que una pieza es intercambiada, el sistema espera dos unidades de tiempo, que son marcadas por la variable C que indica que durante ese tiempo no es evaluada la inactividad de la estación de trabajo. El proceso termina una vez que se encuentre el stock vacío.

5.3 Observaciones

El problema de secuencia y en particular el de secuencia de automóviles se presenta como un problema NP hard [Kis, 2003] dado el número de posibles configuraciones que se presentan como permutaciones de los elementos en la secuencia: n!.

A través del modelo propuesto, el número de intercambios, llámense permutaciones, se reduce, ya que los intercambios se realizan únicamente con los elementos que no han sido consumidos, con lo cual cada vez que el reloj avanza, se reduce el número de elementos candidatos al intercambio:

A partir de TG, la evaluación de las líneas de producción inicia, para cada tiempo mayor a TG, se establece la variable t' cuyo dominio es [TG, max{fpy(x_{i,j})}]:

$$t' \leq \text{Max}\{|V_A(j)| * zy(j), j = 1, \dots, n\}.$$

Sea $H = \{x_{i,j} \ni ipy_j(x_{i,j}) > t'\}$ entonces el número máximo de intercambios está dado por $|H|!$, en el tiempo $t' = t + zy_j$, al menos un elemento del tipo j habrá sido consumido, y por lo tanto disminuido el número de intercambios.

Se define $H_{inicial} = \{x_{i,j} \ni ipy_j(x_{i,j}) > TG\}$, como el número de elementos en la secuencia, en esta declaración, entonces $|H_{t'}| \leq |H_{inicial}|!$.

La propuesta en el modelo es conocer el error en el momento en que este aparece y repararlo, avanzar mientras se reparan las decisiones de configuración, puesto que encontrar la configuración que evite los stock cero en el inicio de la producción representa las permutaciones de todo el conjunto.

Capítulo 6

CASO DE ESTUDIO Y RESULTADOS

6.1 Representación del Sistema. Método de Intercambio

El algoritmo propuesto para resolver el problema de secuenciación descrito en el capítulo anterior, se aplica a un ejemplo académico, de 47 unidades a producir más 10 elementos a la mitad del proceso, la información y desarrollo se describe a continuación.

Se producen 5 tipos de automóviles $j=1, 2, 3, 4, 5$ tipos de automóviles:

$m_1= 10$ $m_2= 8$ $m_3= 10$ $m_4= 9$ $m_5= 10$	$M = \text{Max}\{ V_j = m_j; j = 1, \dots, 5\}$ $M=10$
---	--

$V =$

Tipo1	Tipo2	Tipo3	Tipo4	Tipo5
x1,1	x1,2	x1,3	x1,4	x1,5
x2,1	x2,2	x2,3	x2,4	x2,5
x3,1	x3,2	x3,3	x3,4	x3,5
x4,1	x4,2	x4,3	x4,4	x4,5
x5,1	x5,2	x5,3	x5,4	x5,5
x6,1	x6,2	x6,3	x6,4	x6,5
x7,1	x7,2	x7,3	x7,4	x7,5
x8,1	x8,2	x8,3	x8,4	x8,5
x9,1	x9,2	x9,3	x9,4	x9,5
x10,1	x10,2	x10,3	x10,4	x10,5

\Rightarrow

$f(V) =$

Tipo1	Tipo2	Tipo3	Tipo4	Tipo5
31	3	16	12	45
21	18	20	39	10
24	40	34	38	22
13	36	4	25	8
15	17	26	46	47
35	44	19	42	29
1	41	14	9	2
33	5	27	30	23
37	43	28	0	6
32	0	11	0	7

$= L$

Número total de unidades que inician en el proceso común es $|V| = z = 47$.

Secuencia 1 h_s :	$h1$	$x7,1$	$h11$	$x10,3$	$h21$	$x2,1$	$h31$	$x1,1$	$h41$	$x7,2$
	$h2$	$x7,5$	$h12$	$x1,4$	$h22$	$x3,5$	$h32$	$x10,1$	$h42$	$x6,4$
	$h3$	$x1,2$	$h13$	$x4,1$	$h23$	$x8,5$	$h33$	$x8,1$	$h43$	$x9,2$
	$h4$	$x4,3$	$h14$	$x7,3$	$h24$	$x3,1$	$h34$	$x3,3$	$h44$	$x6,2$
	$h5$	$x8,2$	$h15$	$x5,1$	$h25$	$x4,4$	$h35$	$x6,1$	$h45$	$x1,5$
	$h6$	$x9,5$	$h16$	$x1,3$	$h26$	$x5,3$	$h36$	$x4,2$	$h46$	$x5,4$
	$h7$	$x10,5$	$h17$	$x5,2$	$h27$	$x8,3$	$h37$	$x9,1$	$h47$	$x5,5$
	$h8$	$x4,5$	$h18$	$x2,2$	$h28$	$x9,3$	$h38$	$x3,4$		
	$h9$	$x7,4$	$h19$	$x6,3$	$h29$	$x6,5$	$h39$	$x2,4$		
	$h10$	$x2,5$	$h20$	$x2,3$	$h30$	$x8,4$	$h40$	$x3,2$		

Sea $zx=2$, el tiempo en que tarda la estación X (proceso común) en procesar los vehículos, entonces se calcula los tiempos de inicio y fin para cada vehículo en la matriz $x_{i,j} = (h_s, ipx, fpx)$:

$$ipx_s(x_{i,j}) := \begin{cases} 1 & s = 1 \quad [f(x_{i,j}) = k_{i,j} \ni h_1 = k_{i,j}] \\ fpx_{s-1}(x_{i,j}) + 1 & s > 1 \quad [f(x_{i,j}) = k_{i,j} \ni h_s = k_{i,j} \quad y \quad s \in \{2, \text{Max}(h_t) = z'\}] \end{cases}$$

$$fpx_s(x_{i,j}) := ipx_s + zx$$

$x1,1=(h31, 61, 62)$	$x1,2=(h3, 5, 6)$	$x1,3=(h16, 31, 32)$	$x1,4=(h12, 23, 24)$	$x1,5=(h45, 89, 90)$
$x2,1=(h21, 41, 42)$	$x2,2=(h18, 35, 36)$	$x2,3=(h20, 39, 40)$	$x2,4=(h39, 77, 78)$	$x2,5=(h10, 19, 20)$
$x3,1=(h24, 47, 48)$	$x3,2=(h40, 79, 80)$	$x3,3=(h34, 67, 68)$	$x3,4=(h38, 75, 76)$	$x3,5=(h22, 43, 44)$
$x4,1=(h13, 25, 26)$	$x4,2=(h36, 71, 72)$	$x4,3=(h4, 7, 8)$	$x4,4=(h25, 49, 50)$	$x4,5=(h8, 15, 16)$
$x5,1=(h15, 29, 30)$	$x5,2=(h17, 33, 34)$	$x5,3=(h26, 51, 52)$	$x5,4=(h46, 91, 92)$	$x5,5=(h47, 93, 94)$
$x6,1=(h35, 69, 70)$	$x6,2=(h44, 87, 88)$	$x6,3=(h19, 37, 38)$	$x6,4=(h42, 83, 84)$	$x6,5=(h29, 57, 58)$
$x7,1=(h1, 1, 2)$	$x7,2=(h41, 81, 82)$	$x7,3=(h14, 27, 28)$	$x7,4=(h9, 17, 18)$	$x7,5=(h2, 3, 4)$
$x8,1=(h33, 65, 66)$	$x8,2=(h5, 9, 10)$	$x8,3=(h27, 53, 54)$	$x8,4=(h30, 59, 60)$	$x8,5=(h23, 45, 46)$
$x9,1=(h37, 73, 74)$	$x9,2=(h43, 85, 86)$	$x9,3=(h28, 55, 56)$		$x9,5=(h6, 11, 12)$
$x10,1=(h32, 63, 64)$		$x10,3=(h11, 21, 22)$		$x10,5=(h7, 13, 14)$

Sea $V_A = \{x^a_{i,j} \quad 0 < i \leq ma_j \quad 0 < j \leq n\}$ y $|V_A| = 10$,

$$V_A = \{x^a_{1,1}, x^a_{2,1}, x^a_{1,2}, x^a_{2,2}, x^a_{1,3}, x^a_{2,3}, x^a_{1,4}, x^a_{2,4}, x^a_{1,5}, x^a_{2,5}\} \Rightarrow$$

h_s : Secuencia del día anterior:

$x^{a1,1}=(h8, 0, 0)$	$x^{a1,2}=(h1, 0, 0)$	$x^{a1,3}=(h6, 0, 0)$	$x^{a1,4}=(h9, 0, 0)$	$x^{a1,5}=(h7, 0, 0)$
$x^{a2,1}=(h3, 0, 0)$	$x^{a2,2}=(h4, 0, 0)$	$x^{a2,3}=(h5, 0, 0)$	$x^{a2,4}=(h10, 0, 0)$	$x^{a2,5}=(h2, 0, 0)$

Luego entonces, para el conjunto total se tiene la nueva secuencia:

$$x_{i,j} = (h_{2s}, ipx, fpx), \text{ donde } 2s=1, \dots, r, 1+r, \dots, z \text{ y } x_{i,j} \in VUV_A = V_T$$

h_{2s} : Secuencia incorporando todos los automóviles:

x1,1=(h41, 91, 93)	x1,2=(h13, 7, 9)	x1,3=(h26, 46, 48)	x1,4=(h22, 34, 36)	x1,5=(h55, 133, 135)
x2,1=(h31, 61, 63)	x2,2=(h28, 52, 54)	x2,3=(h30, 58, 60)	x2,4=(h49, 115, 117)	x2,5=(h20, 28, 30)
x3,1=(h34, 70, 72)	x3,2=(h50, 118, 120)	x3,3=(h44, 100, 102)	x3,4=(h48, 112, 114)	x3,5=(h32, 64, 66)
x4,1=(h23, 37, 39)	x4,2=(h46, 106, 108)	x4,3=(h14, 10, 12)	x4,4=(h35, 73, 75)	x4,5=(h18, 22, 24)
x5,1=(h25, 43, 45)	x5,2=(h27, 49, 51)	x5,3=(h36, 76, 78)	x5,4=(h56, 136, 138)	x5,5=(h57, 139, 141)
x6,1=(h45, 103, 105)	x6,2=(h54, 130, 132)	x6,3=(h29, 55, 57)	x6,4=(h52, 124, 126)	x6,5=(h39, 85, 87)
x7,1=(h11, 1, 3)	x7,2=(h51, 121, 123)	x7,3=(h24, 40, 42)	x7,4=(h19, 25, 27)	x7,5=(h12, 4, 6)
x8,1=(h43, 97, 99)	x8,2=(h15, 13, 15)	x8,3=(h37, 79, 81)	x8,4=(h40, 88, 90)	x8,5=(h33, 67, 69)
x9,1=(h47, 109, 111)	x9,2=(h53, 127, 129)	x9,3=(h38, 82, 84)		x9,5=(h16, 16, 18)
x10,1=(h42, 94, 96)		x10,3=(h21, 31, 33)		x10,5=(h17, 19, 21)
x11,1=(h8, 0, 0)	x11,2=(h1, 0, 0)	x11,3=(h6, 0, 0)	x11,4=(h9, 0, 0)	x11,5=(h7, 0, 0)
x12,1=(h3, 0, 0)	x12,2=(h4, 0, 0)	x12,3=(h5, 0, 0)	x12,4=(h10, 0, 0)	x12,5=(h2, 0, 0)

Clasificación por tipo de pieza dentro de la secuencia s2: $Tr(j) = \{h_{2s} \ni h_{2s} := k_{i,j} \quad i = 1, \dots, m_j\}$

$h'_{s(j)}$: Secuencia en la línea de producción j:

<i>h'1(1) x9,1</i>	<i>h'1(2) x11,2</i>	<i>h'1(3) x12,3</i>	<i>h'1(4) x11,4</i>	<i>h'1(5) x12,5</i>
<i>h'2(1) x6,1</i>	<i>h'2(2) x12,2</i>	<i>h'2(3) x11,3</i>	<i>h'2(4) x12,4</i>	<i>h'2(5) x11,5</i>
<i>h'3(1) x3,1</i>	<i>h'3(2) x1,2</i>	<i>h'3(3) x4,3</i>	<i>h'3(4) x7,4</i>	<i>h'3(5) x7,5</i>
<i>h'4(1) x4,1</i>	<i>h'4(2) x8,2</i>	<i>h'4(3) x10,3</i>	<i>h'4(4) x1,4</i>	<i>h'4(5) x9,5</i>
<i>h'5(1) x5,1</i>	<i>h'5(2) x5,2</i>	<i>h'5(3) x7,3</i>	<i>h'5(4) x4,4</i>	<i>h'5(5) x10,5</i>
<i>h'6(1) x11,1</i>	<i>h'6(2) x2,2</i>	<i>h'6(3) x1,3</i>	<i>h'6(4) x8,4</i>	<i>h'6(5) x4,5</i>
<i>h'7(1) x7,1</i>	<i>h'7(2) x4,2</i>	<i>h'7(3) x6,3</i>	<i>h'7(4) x8,4</i>	<i>h'7(5) x2,5</i>
<i>h'8(1) x12,1</i>	<i>h'8(2) x3,2</i>	<i>h'8(3) x2,3</i>	<i>h'7(4) x3,4</i>	<i>h'8(5) x3,5</i>
<i>h'9(1) x8,1</i>	<i>h'9(2) x7,2</i>	<i>h'9(3) x5,3</i>	<i>h'8(4) x2,4</i>	<i>h'9(5) x8,5</i>
<i>h'10(1) x1,1</i>	<i>h'10(2) x9,2</i>	<i>h'10(3) x8,3</i>	<i>h'9(4) x6,4</i>	<i>h'10(5) x6,5</i>
<i>h'11(1) x2,1</i>	<i>h'11(2) x6,2</i>	<i>h'11(3) x9,3</i>	<i>h'10(4) x5,4</i>	<i>h'11(5) x1,5</i>
<i>h'12(1) x10,1</i>		<i>h'12(3) x3,3</i>		<i>h'12(5) x5,5</i>

Sea:

$$zy(1) = 10; \quad zy(2) = 14 \quad zy(3) = 12 \quad zy(4) = 13 \quad zy(5) = 11$$

Considerando los tiempos de procesamiento anteriores y la secuencia $h'_{s(j)}$, se obtiene los valores para las variables $ipy_{s(j)}(x_{i,j})$, $fpy_{s(j)}(x_{i,j})$:

$$ipy_{s(j)}(x_{i,j}) := \begin{cases} 1 & s = 1 \\ fpy_{s-1(j)} + 1 & s > 1 \\ \max\{fpy_{s-1(j)}, fpx_s\} & s > 1 \end{cases} \quad \begin{matrix} fpx_{s(j)} = 0 \\ fpx_{s(j)} \neq 0 \end{matrix}$$

$$fpy_{s(j)} := ipy_{s(j)} + zy(j) \quad \Rightarrow \quad x_{i,j} = (h_{2s}, ipx, fpx, h'_{s(j)}, ipy_j, fpy_j)$$

x9,1=[h2.3, 0, 0, h'1(1), 138, 148]
x6,1=[h2.8, 0, 0, h'2(1), 127, 137]
x3,1=[h2.11, 1, 3, h'3(1), 75, 85]
x4,1=[h2.23, 37, 39, h'4(1), 40, 50]
x5,1=[h2.25, 43, 45, h'5(1), 51, 61]
x11,1=[h2.31, 61, 63, h'6(1), 12, 22]
x7,1=[h2.34, 70, 72, h'7(1), 23, 33]
x12,1=[h2.41, 91, 93, h'8(1), 1, 11]
x8,1=[h2.42, 94, 96, h'9(1), 116, 126]
x1,1=[h2.43, 97, 99, h'10(1), 94, 104]
x2,1=[h2.45, 103, 105, h'11(1), 64, 74]
x10,1=[h2.47, 109, 111, h'12(1), 105, 115]

x11,4=[h2.9, 0, 0, h'1(4), 1, 14]
x12,4=[h2.10, 0, 0, h'2(4), 15, 28]
x7,4=[h2.19, 25, 27, h'3(4), 29, 42]
x1,4=[h2.22, 34, 36, h'4(4), 43, 56]
x4,4=[h2.35, 73, 75, h'5(4), 76, 89]
x8,4=[h2.40, 88, 90, h'6(4), 90, 103]
x3,4=[h2.48, 112, 114, h'7(4), 115, 128]
x2,4=[h2.49, 115, 117, h'8(4), 129, 142]
x6,4=[h2.52, 124, 126, h'9(4), 143, 156]
x5,4=[h2.56, 136, 138, h'10(4), 157, 170]

x11,2=[h2.1, 0, 0, h'1(2), 1, 15]
x12,2=[h2.4, 0, 0, h'2(2), 16, 30]
x1,2=[h2.13, 7, 9, h'3(2), 31, 45]
x8,2=[h2.15, 13, 15, h'4(2), 46, 60]
x5,2=[h2.27, 49, 51, h'5(2), 61, 75]
x2,2=[h2.28, 52, 54, h'6(2), 76, 90]
x4,2=[h2.46, 106, 108, h'7(2), 109, 123]
x3,2=[h2.50, 118, 120, h'8(2), 124, 138]
x7,2=[h2.51, 121, 123, h'9(2), 139, 153]
x9,2=[h2.53, 127, 129, h'10(2), 154, 168]
x6,2=[h2.54, 130, 132, h'11(2), 169, 183]

x12,5=[h2.2, 0, 0, h'1(5), 1, 12]
x11,5=[h2.7, 0, 0, h'2(5), 13, 24]
x7,5=[h2.12, 4, 6, h'3(5), 25, 36]
x9,5=[h2.16, 16, 18, h'4(5), 37, 48]
x10,5=[h2.17, 19, 21, h'5(5), 49, 60]
x4,5=[h2.18, 22, 24, h'6(5), 61, 72]
x2,5=[h2.20, 28, 30, h'7(5), 73, 84]
x3,5=[h2.32, 64, 66, h'8(5), 85, 96]
x8,5=[h2.33, 67, 69, h'9(5), 97, 108]
x6,5=[h2.39, 85, 87, h'10(5), 109, 120]
x1,5=[h2.55, 133, 135, h'11(5), 136, 147]
x5,5=[h2.57, 139, 141, h'12(5), 148, 159]

x12,3=[h2.5, 0, 0, h'1(3), 1, 13]
x11,3=[h2.6, 0, 0, h'2(3), 14, 26]
x4,3=[h2.14, 10, 12, h'3(3), 27, 39]
x10,3=[h2.21, 31, 33, h'4(3), 40, 52]
x7,3=[h2.24, 40, 42, h'5(3), 53, 65]
x1,3=[h2.26, 46, 48, h'6(3), 66, 78]
x6,3=[h2.29, 55, 57, h'7(3), 79, 91]
x2,3=[h2.30, 58, 60, h'8(3), 92, 104]
x5,3=[h2.36, 76, 78, h'9(3), 105, 117]
x8,3=[h2.37, 79, 81, h'10(3), 118, 130]
x9,3=[h2.38, 82, 84, h'11(3), 131, 143]
x3,3=[h2.44, 100, 102, h'12(3), 144, 156]

Antes de evaluar las colar en el tiempo t , hay que tomar en cuenta el periodo de “gracia” establecido:

$$TG = \text{Max}\{|V_A(j)| * z_Y(j), j = 1, \dots, n\} \Rightarrow TG = 28$$

Lo que refiere que el sistema será evaluado a partir de $t=28$, siempre y cuando que haya llegado al menos una pieza para cada línea al proceso Y, para este ejemplo $t=52$, como se marca en el siguiente cuadro 6.1.

6.Caso de Estudio

Tiempo	Tipo1	Tipo2	Tipo3	Tipo4	Tipo5	^Caída a cero	TipoX=0	T1-Cola	T2-Cola	T3-Cola	T4-Cola	T5-Cola	T1-tiempo	T2-tiempo	T3-tiempo	T4-tiempo	T5-tiempo	Max (tiempo mayor de retardo)	TipoX=Max
1	0	0	0	0	0	nc	Tipo1	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
2	1021	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
3	1021	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
4	1021	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
5	1021	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
6	1021	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
7	1021	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
8	1021	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
9	1021	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
10	1021	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
11	1021	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
12	1011	1012	1023	1014	1025	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
13	1011	1012	1023	1014	1015	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
14	1011	1012	1013	1014	1015	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
15	1011	1012	1013	1024	1015	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
16	1011	1022	1013	1024	1015	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
17	1011	1022	1013	1024	1015	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
18	1011	1022	1013	1024	1015	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
19	1011	1022	1013	1024	1015	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
20	1011	1022	1013	1024	1015	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
21	1011	1022	1013	1024	1015	nc	en proceso	TG	TG	TG	TG	TG	nc	nc	nc	nc	nc	nc	nc
22	1011	1022	1013	1024	1015	nc	en proceso	TG	TG	TG	TG	TG	3	nc	nc	nc	nc	33	nc
23	0071	1022	1013	1024	1015	nc	en proceso	TG	TG	TG	TG	TG	3	nc	nc	nc	nc	33	nc
24	0071	1022	1013	1024	1015	nc	en proceso	TG	TG	TG	TG	TG	3	nc	nc	nc	nc	33	nc
25	0071	1022	1013	1024	0075	nc	en proceso	TG	TG	TG	TG	TG	3	nc	nc	nc	nc	33	nc
26	0071	1022	1013	1024	0075	nc	en proceso	TG	TG	TG	TG	TG	3	nc	nc	nc	nc	33	nc
27	0071	1022	0043	1024	0075	nc	en proceso	TG	TG	TG	TG	TG	3	nc	nc	nc	nc	33	nc
28	0071	1022	0043	1024	0075	nc	en proceso	TG	TG	TG	TG	1	3	nc	nc	nc	13	33	nc
29	0071	1022	0043	0074	0075	nc	en proceso	TG	TG	TG	0	3	nc	nc	nc	0	33	nc	nc
30	0071	1022	0043	0074	0075	nc	en proceso	TG	TG	TG	0	3	nc	nc	nc	0	33	nc	nc
31	0071	0012	0043	0074	0075	nc	en proceso	TG	TG	TG	0	4	nc	nc	nc	0	44	nc	nc
32	0071	0012	0043	0074	0075	nc	en proceso	TG	TG	TG	0	4	nc	nc	nc	0	44	nc	nc
33	0071	0012	0043	0074	0075	nc	en proceso	TG	TG	TG	0	4	nc	nc	nc	0	44	nc	nc
34	0	0012	0043	0074	0075	nc	Tipo1	TG	TG	1	0	4	nc	nc	12	0	44	nc	nc
35	0	0012	0043	0074	0075	nc	Tipo1	TG	TG	1	0	4	nc	nc	12	0	44	nc	nc
36	0	0012	0043	0074	0075	nc	Tipo1	TG	TG	1	0	4	nc	nc	12	0	44	nc	nc
37	0	0012	0043	0074	0095	nc	Tipo1	TG	TG	1	1	3	nc	nc	12	13	33	nc	nc
38	0	0012	0043	0074	0095	nc	Tipo1	TG	TG	1	1	3	nc	nc	12	13	33	nc	nc
39	0	0012	0043	0074	0095	nc	Tipo1	TG	TG	1	1	3	nc	nc	12	13	33	nc	nc
40	0041	0012	0103	0074	0095	nc	en proceso	0	TG	0	1	3	0	nc	0	13	33	nc	nc
41	0041	0012	0103	0074	0095	nc	en proceso	0	TG	0	1	3	0	nc	0	13	33	nc	nc
42	0041	0012	0103	0074	0095	nc	en proceso	0	TG	0	1	3	0	nc	0	13	33	nc	nc
43	0041	0012	0103	0014	0095	nc	en proceso	0	TG	1	0	3	0	nc	12	0	33	nc	nc
44	0041	0012	0103	0014	0095	nc	en proceso	0	TG	1	0	3	0	nc	12	0	33	nc	nc
45	0041	0012	0103	0014	0095	nc	en proceso	0	TG	1	0	3	0	nc	12	0	33	nc	nc
46	0041	0082	0103	0014	0095	nc	en proceso	1	TG	1	0	3	10	nc	12	0	33	nc	nc
47	0041	0082	0103	0014	0095	nc	en proceso	1	TG	1	0	3	10	nc	12	0	33	nc	nc
48	0041	0082	0103	0014	0095	nc	en proceso	1	TG	1	0	3	10	nc	12	0	33	nc	nc
49	0041	0082	0103	0014	0105	nc	en proceso	1	TG	2	0	2	10	nc	24	0	22	nc	nc
50	0041	0082	0103	0014	0105	nc	en proceso	1	TG	2	0	2	10	nc	24	0	22	nc	nc
51	0051	0082	0103	0014	0105	nc	en proceso	0	TG	2	0	2	0	nc	24	0	22	nc	nc
52	0051	0082	0103	0014	0105	ocupado	en proceso	0	1	2	0	2	0	14	24	0	22	24	Tipo3
53	0051	0082	0073	0014	0105	ocupado	en proceso	0	1	1	0	2	0	14	12	0	22	22	Tipo5
54	0051	0082	0073	0014	0105	ocupado	en proceso	0	1	1	0	2	0	14	12	0	22	22	Tipo5
55	0051	0082	0073	0014	0105	ocupado	en proceso	0	2	1	0	2	0	28	12	0	22	28	Tipo2
56	0051	0082	0073	0014	0105	ocupado	en proceso	0	2	1	0	2	0	28	12	0	22	28	Tipo2
57	0051	0082	0073	0	0105	cero	Tipo4	0	2	1	0	2	0	28	12	0	22	28	Tipo2
58	0051	0082	0073	0	0105	cero	Tipo4	0	2	2	0	2	0	28	24	0	22	28	Tipo2
59	0051	0082	0073	0	0105	cero	Tipo4	0	2	2	0	2	0	28	24	0	22	28	Tipo2

Cuadro 6.1 Caída a cero.

A continuación se muestran los primeros elementos con sus secuencias: general h2 y por tipo para cada una de las líneas de producción h's(1), h's(2), h's(3), h's(4), h's(5), con los tiempos de proceso para cada vehículo:

h's(1)	h2	Posición	Tipo	Inicio PX	Fin PX	Inicio PY	Fin PY	Auxx (TG)
1	3	0111	Tipo1	0	0	1	11	TG
2	8	0121	Tipo1	0	0	12	22	TG
3	11	0071	Tipo1	1	3	23	33	11
4	23	0041	Tipo1	37	39	40	50	23
5	25	0051	Tipo1	43	45	51	61	25
6	31	0021	Tipo1	61	63	64	74	31
h's(2)	h2	Posición	Tipo	Inicio PX	Fin PX	Inicio PY	Fin PY	Auxx (TG)
1	1	0112	Tipo2	0	0	1	15	TG
2	4	0122	Tipo2	0	0	16	30	TG
3	13	0012	Tipo2	7	9	31	45	13
4	15	0082	Tipo2	13	15	46	60	15

h's(3)	h2	Posición	Tipo	Inicio PX	Fin PX	Inicio PY	Fin PY	Auxx (TG)
1	5	0113	Tipo3	0	0	1	13	TG
2	6	0123	Tipo3	0	0	14	26	TG
3	14	0043	Tipo3	10	12	27	39	14
4	21	0103	Tipo3	31	33	40	52	21
5	24	0073	Tipo3	40	42	53	65	24
6	26	0013	Tipo3	46	48	66	78	26
h's(4)	h2	Posición	Tipo	Inicio PX	Fin PX	Inicio PY	Fin PY	Auxx (TG)
1	9	0114	Tipo4	0	0	1	14	TG
2	10	0124	Tipo4	0	0	15	28	10
3	19	0074	Tipo4	25	27	29	42	19
4	22	0014	Tipo4	34	36	43	56	22
5	35	0044	Tipo4	73	75	76	89	35
h's(5)	h2	Posición	Tipo	Inicio PX	Fin PX	Inicio PY	Fin PY	Auxx (TG)
1	2	0115	Tipo5	0	0	1	12	TG
2	7	0125	Tipo5	0	0	13	24	TG
3	12	0075	Tipo5	4	6	25	36	12
4	16	0095	Tipo5	16	18	37	48	16
5	17	0105	Tipo5	19	21	49	60	17
6	18	0045	Tipo5	22	24	61	72	18

Para $t > Tg$ se tiene:

Estación ocupada	$\exists x_{i,j} \ni ipy(x_{i,j}) \leq t \leq fpy(x_{i,j})$
Piezas en cola	$\exists x_{i,j} := h'_{s(j)}, x_{i',j'} := h'_{s(j)+1} \ni ipy(x_{i,j}) \leq t \leq fpy(x_{i,j})$ $fpx(x_{i',j'}) < fpy(x_{i,j})$
Línea vacía	$\exists x_{i,j} := h'_{s(j)}, x_{i',j'} := h'_{s(j)+1} \ni fpy(x_{i,j}) < t$ $ipy(x_{i',j'}) > t$
Stock Vacío	e.o.c.

Entonces en $t=52$, el primer punto de evaluación, todas las líneas se encuentran trabajando; sin embargo, en $t=57$, el esquema nos muestra la caída a cero para la estación de trabajo que produce el tipo 4; la secuencia inicial no ha sido la óptima, una línea de producción se encuentra inactiva por falta de piezas que procesar.

Se Identifica la unidad a intercambiar de acuerdo a la línea de espera más “tardada” por aquella pieza cuya línea de espera se encuentra vacía.

Identificación de elementos a intercambiar

La línea que ha caído a cero corresponde al Tipo 4, para $t=10$ en dicha línea existe $x_{4,4}$ ($i=4, j^*=4$) la pieza que no llega a tiempo provocando la caída a cero, la información correspondiente a esta entrada de la matriz, es el vector:

$$x_{4,4}=[h2.35, 73, 75, h'5(4), 76, 89]$$

El elemento de la matriz, cuya secuencia es previa en esta línea es $x_{1,4}$:

$$x_{1,4}=[h2.22, 34, 36, h'4(4), 43, 56]$$

$$\exists x_{i,j} := h'_{s(j)}, x_{i',j'} := h'_{s(j)+1} \Rightarrow \begin{cases} fpy(x_{i,j}) < t \\ ipy(x_{i',j'}) > t \end{cases}$$

Se evalúan las piezas en el tiempo mismo tiempo t=57 para las diferentes líneas identificando la línea más “tardada”, (cuadro 6.2) ésta es determinada en base al tamaño de la cola y el tiempo en que tarda el proceso en cada línea, para el ejemplo la línea más tardada es la correspondiente al Tipo2: hay dos elementos en espera y de acuerdo con el tiempo de proceso en esta línea, es la línea más tardada con 28 unidades de tiempo en espera.

Tiempo	Tipo1	Tipo2	Tipo3	Tipo4	Tipo5	^Caída a cero	TipoX=0	T1-Cola	T2-Cola	T3-Cola	T4-Cola	T5-Cola	T1-tiempo	T2-tiempo	T3-tiempo	T4-tiempo	T5-tiempo	Max (tiempo mayor de retardo)	TipoX=Max
52	0051	0082	0103	0014	0105	ocupado	en proceso	0	1	2	0	2	0	14	24	0	22	24	Tipo3
53	0051	0082	0073	0014	0105	ocupado	en proceso	0	1	1	0	2	0	14	12	0	22	22	Tipo5
54	0051	0082	0073	0014	0105	ocupado	en proceso	0	1	1	0	2	0	14	12	0	22	22	Tipo5
55	0051	0082	0073	0014	0105	ocupado	en proceso	0	2	1	0	2	0	28	12	0	22	28	Tipo2
56	0051	0082	0073	0014	0105	ocupado	en proceso	0	2	1	0	2	0	28	12	0	22	28	Tipo2
57	0051	0082	0073	0	0105	cero	Tipo4	0	2	1	0	2	0	28	12	0	22	28	Tipo2
58	0051	0082	0073	0	0105	cero	Tipo4	0	2	2	0	2	0	28	24	0	22	28	Tipo2
59	0051	0082	0073	0	0105	cero	Tipo4	0	2	2	0	2	0	28	24	0	22	28	Tipo2
60	0051	0082	0073	0	0105	cero	Tipo4	0	2	2	0	2	0	28	24	0	22	28	Tipo2
61	0051	0052	0073	0	0045	cero	Tipo4	0	1	3	0	1	0	14	36	0	11	36	Tipo3
62	0	0052	0073	0	0045	cero	Tipo1	0	1	3	0	1	0	14	36	0	11	36	Tipo3
63	0	0052	0073	0	0045	cero	Tipo1	0	1	3	0	1	0	14	36	0	11	36	Tipo3
64	0021	0052	0073	0	0045	cero	Tipo4	0	1	3	0	1	0	14	36	0	11	36	Tipo3

Cuadro 6.2. Identificación de la línea “más” tardada.

La información correspondiente a esta línea en la evaluación:

$$x_{8,2}=[h_{2.15}, 13, 15, h'_{4(2)}, 46, 60]$$

La siguiente pieza en entrar al proceso Y es en esta línea es:

$$x_{5,2}=[h_{2.27}, 49, 51, h'_{5(2)}, 61, 75]$$

El lugar de la secuencia de $x_{4,4}$ es $h_{2.35}$, y para $x_{5,2}$ es $h_{2.27}$ entonces se intercambian sus lugares en la secuencia, evitando así que la línea 4 caiga a cero:

$$t^* = (x_{i,j^c}, h_{2s}, ipx, fpx, h'_{s(j^c)}, ipy, fpy), \Rightarrow ipy(x_{i,j^c}) \leq t^* \leq fpy(x_{i,j^c})$$

$$t^* = 52 \Rightarrow (x_{i=4, j^c=4}, h_{2s=35}, ipx = 73, fpx = 75, h'_{s=5(j^c=4)}, ipy = 76, fpy = 89)$$

Procedimiento de Intercambio

Para $t^* \exists j^* := \text{caída a cero}$ y $j^c := \text{cola más tardada}$,

$$t^* = (x_{i^*, j^*}, h_{2s} [x_{i^*, j^*}], ipx, fpx, h'_{s(j^*)} [x_{i^*, j^*}], ipy, fpy) \quad i^*=4, j^*=4$$

$$t^* = (x_{i^c, j^c}, h_{2s} [x_{i^c, j^c}], ipx, fpx, h'_{s(j^c)} [x_{i^c, j^c}], ipy, fpy) \quad i^c=5, j^c=2$$

Se intercambia la secuencia del vehículo $x_{i^*, j^*} := h_{2s}$, con la secuencia de $x_{i^c, j^c} := h_{2s}$.

Al ir avanzando el proceso, es necesario indicar cuales piezas han sido consumidas o están a punto de finalizar su proceso, en cada intercambio se evalúan la secuencia de las unidades, marcando aquellas que han salido del sistema:

$$sl_t[x_{ij}] = \begin{cases} M & \text{ipy}(x_{ij}) > t \\ h_{2s}(x_{ij}) & \text{e.o.c} \end{cases}$$

Cada vez que una pieza es intercambiada, el sistema espera al menos a que esa pieza sea consumida, para continuar con la evaluación de las colas y la caída a cero en las estaciones.

Entonces a manera de resumen, se tiene en la siguiente figura 6.1, los elementos propuestos para el intercambio, y la representación matricial previo al intercambio y después de haberse realizado el mismo, tercera matriz nos representa las unidades que han sido consumidas al alcanzar el tiempo en el cual se ha realizado el intercambio.

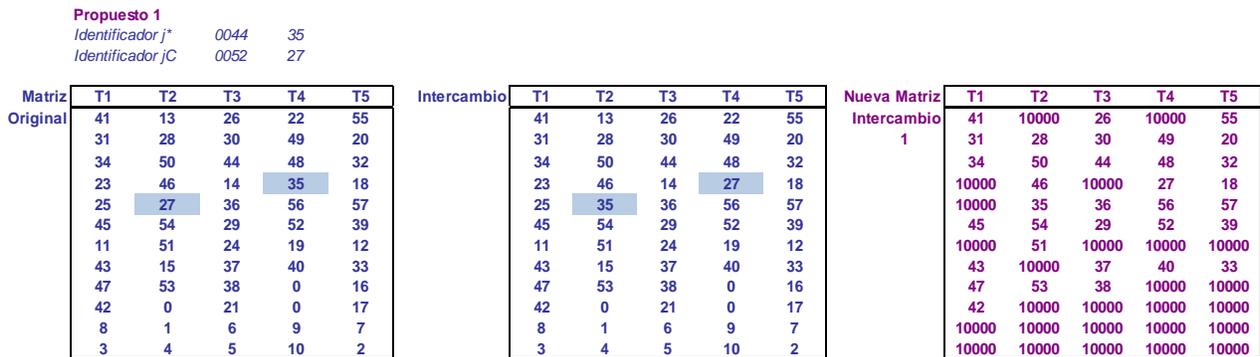


Figura 6.1 Elementos propuestos para el intercambio.

Evaluación de colas y nueva caída a cero

La siguiente caída a cero es identificada en t=62.

Después del intercambio se indica las unidades que se han intercambiado “C”, las cuales ya no son candidatas a un nuevo cambio, así el conjunto de piezas candidatas al nuevo intercambio serán aquellas cuyo tiempo de inicio para el proceso Y es posterior a la caída, siempre y cuando no se hayan intercambiado en la caída anterior. Cuadro 6.3.

6.Caso de Estudio

Tiempo	Tipo1	Tipo2	Tipo3	Tipo4	Tipo5	^Caída a cero	TipoX=0	T1-Cola	T2-Cola	T3-Cola	T4-Cola	T5-Cola	T1-tiempo	T2-tiempo	T3-tiempo	T4-tiempo	T5-tiempo	Max (tiempo mayor de retardo)	TipoX=Max	
1	C	C	C	C	C	ocupado	en proceso	0	0	0	0	0	0	0	0	0	0	0	0	←Max (or
54	0051	C	0073	0014	C	ocupado	en proceso	0	0	1	1	1	0	0	12	13	11	13	Tipo4	
55	0051	C	0073	0014	C	ocupado	en proceso	0	0	1	1	1	0	0	12	13	11	13	Tipo4	
56	0051	C	0073	0014	C	ocupado	en proceso	0	0	1	1	1	0	0	12	13	11	13	Tipo4	
57	0051	C	0073	0044	C	ocupado	en proceso	0	0	1	0	1	0	0	12	0	11	12	Tipo3	
58	0051	C	0073	0044	C	ocupado	en proceso	0	0	2	0	1	0	0	24	0	11	24	Tipo3	
59	0051	C	0073	0044	C	ocupado	en proceso	0	0	2	0	1	0	0	24	0	11	24	Tipo3	
60	0051	C	0073	0044	C	ocupado	en proceso	0	0	2	0	1	0	0	24	0	11	24	Tipo3	
61	0051	C	0073	0044	0045	ocupado	en proceso	0	0	3	0	1	0	0	36	0	11	36	Tipo3	
62	0	C	0073	0044	0045	cero	Tipo1	0	0	3	0	1	0	0	36	0	11	36	Tipo3	
63	0	C	0073	0044	0045	cero	Tipo1	0	0	3	0	1	0	0	36	0	11	36	Tipo3	
64	0021	C	0073	0044	0045	ocupado	en proceso	0	0	3	0	1	0	0	36	0	11	36	Tipo3	
65	0021	C	0073	0044	0045	ocupado	en proceso	0	0	3	0	1	0	0	36	0	11	36	Tipo3	
66	0021	C	0013	0044	0045	ocupado	en proceso	0	0	2	0	1	0	0	24	0	11	24	Tipo3	
67	0021	C	0013	0044	0045	ocupado	en proceso	0	0	2	0	2	0	0	24	0	22	24	Tipo3	
68	0021	C	0013	0044	0045	ocupado	en proceso	0	0	2	0	2	0	0	24	0	22	24	Tipo3	

Cuadro 6.3. Intercambios posteriores al inicial.

En los siguientes cuadros muestran las unidades que se encuentran en el conjunto de intercambio:

h2	Posición	Tipo	Inicio PX	Fin PX	Inicio PY	Fin PY	Auxx (TG)
31	0021	Tipo1	61	63	64	74	31
34	0031	Tipo1	70	72	75	85	34
41	0011	Tipo1	91	93	94	104	41
42	0101	Tipo1	94	96	105	115	42
43	0081	Tipo1	97	99	116	126	43
45	0061	Tipo1	103	105	127	137	45
47	0091	Tipo1	109	111	138	148	47
x	x	x	x	x	y	x	x
h2	Posición	Tipo	Inicio PX	Fin PX	Inicio PY	Fin PY	Auxx (TG)
28	0022	Tipo2	52	54	76	90	28
35	0052	Tipo2	73	75	91	105	35
46	0042	Tipo2	106	108	109	123	46
50	0032	Tipo2	118	120	124	138	50
51	0072	Tipo2	121	123	139	153	51
53	0092	Tipo2	127	129	154	168	53
54	0062	Tipo2	130	132	169	183	54
x	x	x	x	x	y	x	x
h2	Posición	Tipo	Inicio PX	Fin PX	Inicio PY	Fin PY	Auxx (TG)
26	0013	Tipo3	46	48	66	78	26
29	0063	Tipo3	55	57	79	91	29
30	0023	Tipo3	58	60	92	104	30
36	0053	Tipo3	76	78	105	117	36
37	0083	Tipo3	79	81	118	130	37
38	0093	Tipo3	82	84	131	143	38
44	0033	Tipo3	100	102	144	156	44
x	x	x	x	x	y	x	x
h2	Posición	Tipo	Inicio PX	Fin PX	Inicio PY	Fin PY	Auxx (TG)
19	0074	Tipo4	25	27	29	42	19
22	0014	Tipo4	34	36	43	56	22
27	0044	Tipo4	49	51	57	70	27
40	0084	Tipo4	88	90	91	104	40
48	0034	Tipo4	112	114	115	128	48
49	0024	Tipo4	115	117	129	142	49
52	0064	Tipo4	124	126	143	156	52
56	0054	Tipo4	136	138	157	170	56
x	x	x	x	x	y	x	x

h2	Posición	Tipo	Inicio PX	Fin PX	Inicio PY	Fin PY	Auxx (TG)
20	0025	Tipo5	28	30	73	84	20
32	0035	Tipo5	64	66	85	96	32
33	0085	Tipo5	67	69	97	108	33
39	0065	Tipo5	85	87	109	120	39
55	0015	Tipo5	133	135	136	147	55
57	0055	Tipo5	139	141	148	159	57
x	x	x	x	x	y	x	x

Así el siguiente intercambio será entre la línea 1, que ha caído a cero y la línea 3, que en este punto es la línea más tardada, las piezas identificadas se muestran a continuación:

Propuesto 2

Identificador j* 0021 31
 Identificador jC 0013 26

T1	T2	T3	T4	T5
41	13	26	22	55
31	28	30	49	20
34	50	44	48	32
23	46	14	27	18
25	35	36	56	57
45	54	29	52	39
11	51	24	19	12
43	15	37	40	33
47	53	38	0	16
42	0	21	0	17
8	1	6	9	7
3	4	5	10	2

Intercambio 2

T1	T2	T3	T4	T5
41	13	31	22	55
26	28	30	49	20
34	50	44	48	32
23	46	14	27	18
25	35	36	56	57
45	54	29	52	39
11	51	24	19	12
43	15	37	40	33
47	53	38	0	16
42	0	21	0	17
8	1	6	9	7
3	4	5	10	2

Nueva Matriz Intercambio 2

T1	T2	T3	T4	T5
41	10000	31	10000	55
26	28	30	49	20
34	50	44	48	32
10000	46	10000	10000	10000
10000	35	36	56	57
45	54	29	52	39
10000	51	10000	10000	10000
43	10000	37	40	33
47	53	38	10000	10000
42	10000	10000	10000	10000
10000	10000	10000	10000	10000
10000	10000	10000	10000	10000

Figura 6.2 Elementos para intercambios posteriores.

Este proceso de evaluación de colas e identificación del estatus de cada estación de trabajo, se repite hasta que la siguiente caída a cero sea presentada por el final de las piezas en stock para todas las líneas, como se muestra en la tabla 6.1.

Intercambios realizados							
Intercambio	Tiempo	Línea caída a cero	Identificador j*	Lugar en la secuencia	Línea de intercambio	Identificador jC	Lugar en la secuencia
1	57	Tipo4	0044	35	Tipo2	0052	27
2	62	Tipo1	0021	31	Tipo3	0013	26
3	71	Tipo4	0084	40	Tipo3	0023	30
4	84	Tipo1	0011	41	Tipo3	0053	36
5	95	Tipo1	0101	42	Tipo3	0033	44
6	96	Tipo1	0081	43	Tipo3	0033	42
7	106	Tipo2	0042	46	Tipo3	0093	38
8	121	Tipo5	0015	55	Tipo3	0033	43

Tabla 6.1 Intercambios.

6.2 Herramienta de Modelado

Basada en el algoritmo propuesto para el problema de secuencia, se ha implementado una herramienta de secuencialización desarrollada en Visual Basic con el objetivo de verificar experimentalmente el algoritmo propuesto. Esta herramienta inicia con una secuencia aleatoria, para posteriormente analizar el estado de cada estación de trabajo y las colas en cada línea especializada, de tal forma que siguiendo el algoritmo, identifica los puntos críticos de inactividad, solucionando el problema en el momento de la ocurrencia del mismo a través de la búsqueda inteligente de piezas a intercambiar.

La aplicación lee la información básica del número de tipos de automóviles con un límite de 5 tipos y el número de elementos en cada tipo, el cual se encuentra limitado a 100; tiempos de procesamiento para los procesos X y Yj, así como información de las unidades que se encuentran a la mitad del proceso el inicio del día.

La información de cada intercambio y la identificación de piezas y tiempo, se genera a partir de la secuencia aleatoria inicial, dada por la misma aplicación.

La herramienta implementada recopila la información (figura 6.3) mínima necesaria para resolver el problema de secuencia bajo el método de intercambio propuesto:

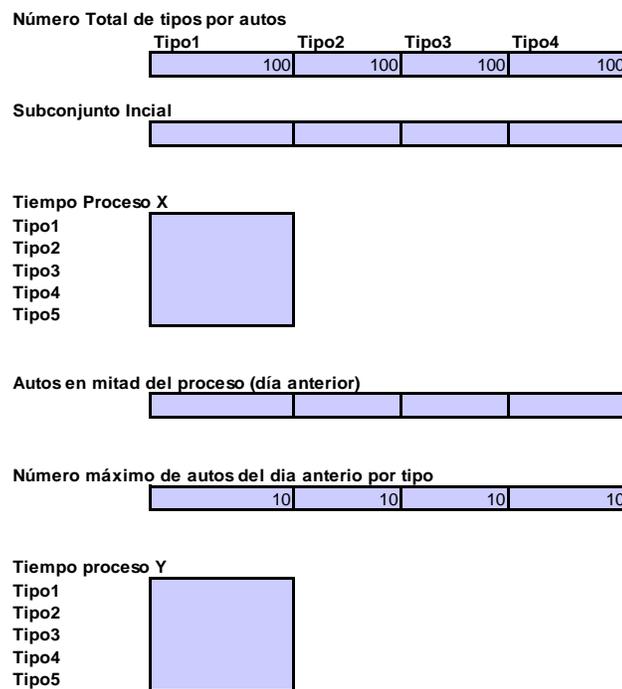


Figura 6.3. Información solicitada en la aplicación.

Esta aplicación está diseñada para operar con cinco tipos de vehículos y con un límite por tipo de 100 elementos.

El usuario debe especificar el número de vehículos dentro de las casillas de Subconjunto Inicial, para que sea establecido el tamaño de la matriz que se manejará en la aplicación.

En el área de Tiempos de proceso X, se debe establecer los tiempos en que tardará el primer proceso o proceso común para todos los vehículo, es importante mencionar que se ha definido el mismo tiempo de proceso para todas las líneas de producción.

Se maneja en el modelo el concepto de “automóviles del día anterior a mitad del proceso” en este apartado se consideran aquellos vehículos que han sido procesados en por la estación correspondiente al proceso X, teniendo como límite superior 10 unidades por tipo de vehículo.

Posteriormente se localiza el área correspondiente a los tiempos de proceso Y, o proceso especializado, aquí el usuario define el tiempo en que cada estación de trabajo especial por tipo tardará en procesar los vehículos que lleguen a cada línea.

El algoritmo lee esta información proporcionada por el usuario y la utiliza para dar inicio a la aplicación; cuando en el proceso identifica la caída a cero para alguna línea de producción y proporciona la información de intercambio, figura 6.4:

El intercambio se realiza entre el y el

Línea caída a cero

Tamaño de la cola
Referencia
Línea
Identificador



Línea mas "tardada" en el momento de la caída

Criterios: Tamaño y tiempo en cola
Referencia
Línea
Identificador



Condiciones:

Identificador j*
Identificador jC



Número de intercambios

Cada vez que se aplique el macro de intercambio registrar uno

Figura 6.4 Identificación de elementos a intercambiar.

Los cambios son registrados en la hoja llamada Registro de Cambios, con la siguiente información:

Número de intercambios

Cada vez que se aplique el macro de intercambio registrar uno

Identificador j* Lugar
0135 45

Identificador jC Lugar
0183 25

Cero a intercambio
Tipo5 Tipo3

El algoritmo puede ejecutarse paso a paso a través de los botones de la izquierda o simular todo el proceso botón "Repetición" cuyo criterio de detención se encuentra dado por los stocks vacíos, figura 6.5.

Tipos de Autos

Número Total de tipos por autos

Tipo1	Tipo2	Tipo3	Tipo4	Tipo5
100	100	100	100	100

Subconjunto Inicial

20	20	20	20	20
----	----	----	----	----

Tiempo Proceso X

Tipo1	2
Tipo2	2
Tipo3	2
Tipo4	2
Tipo5	2

Autos en mitad del proceso (día anterior)

2	2	2	2	2
---	---	---	---	---

Número máximo de autos del día anterior por tipo

10	10	10	10	10
----	----	----	----	----

Tiempo proceso Y

Tipo1	10
Tipo2	14
Tipo3	12
Tipo4	13
Tipo5	11

Alguna línea de producción ha caído a cero antes de terminar con todas las unidades registradas?

Habrà que realizar intercambio de piezas

El intercambio se realiza entre el Tipo5 y el Tipo3

Línea caída a cero

Tamaño de la col: 0
Referencia: 85
Línea: Tipo5
Identificador: 0135

Línea mas "tardada" en el momento de la caída

Criterios:Tamaño: 48
Referencia: 85
Línea: Tipo3
Identificador: 0183

Condiciones:

Identificador j*	0135	45
Identificador jC	0183	25

Información (D100)

Paso 1. Posición Aleatoria (D100)

Paso 2. Tiempo en el Proceso X (D100)

Paso 3. Autos Anteriores (D100)

Paso 4. Tiempo en el Proceso Y (colas) (D100)

Paso 5. Identificación de colas (D100)

Histórico

Paso 5.1. Identificación Resumen (D100)

Repetición

Intercambio (D100)

Revisión de Colas después del Intercambio (D100)

Hora Inicio	Hora Final	Tiempo

Figura 6.5 Pantalla de herramienta de intercambio.

Simulación4. 40 unidades a producir por línea, tabla 6.5

Intercambios realizados. 200 unidades							
Intercambio	Tiempo	Línea caída a cero	Identificador j*	Lugar en la secuencia	Línea de intercambio	Identificador jC	Lugar en la secuencia
1	161	Tipo1	0291	79	Tipo4	0304	47
2	164	Tipo1	0301	84	Tipo2	0262	67
3	167	Tipo1	0041	86	Tipo4	0304	79
4	174	Tipo1	0261	90	Tipo4	0324	82
5	183	Tipo1	0051	92	Tipo2	0262	84
6	189	Tipo1	0201	104	Tipo2	0352	87
7	198	Tipo1	0151	106	Tipo2	0132	91
8	209	Tipo1	0121	118	Tipo2	0242	94
9	219	Tipo1	0251	127	Tipo4	0034	97
10	228	Tipo1	0211	137	Tipo5	0065	99
11	234	Tipo1	0021	139	Tipo5	0235	101
12	240	Tipo1	0401	140	Tipo2	0132	106
13	251	Tipo1	0031	150	Tipo2	0122	108
14	261	Tipo1	0331	153	Tipo3	0323	110
15	267	Tipo1	0241	155	Tipo2	0212	116
16	278	Tipo1	0271	158	Tipo2	0242	118
17	289	Tipo1	0281	165	Tipo2	0302	123
18	300	Tipo1	0371	174	Tipo2	0132	140
19	313	Tipo5	0045	130	Tipo4	0034	127
20	346	Tipo1	0011	176	Tipo3	0313	135
21	349	Tipo3	0273	143	Tipo4	0064	141
22	373	Tipo2	0122	150	Tipo3	0013	148
23	385	Tipo5	0225	157	Tipo2	0282	151
24	399	Tipo4	0074	167	Tipo2	0282	157
25	411	Tipo4	0364	169	Tipo2	0382	159
26	417	Tipo4	0124	172	Tipo2	0252	162
27	426	Tipo4	0164	175	Tipo2	0302	165
28	435	Tipo4	0344	180	Tipo2	0382	169
29	447	Tipo4	0204	183	Tipo2	0252	172
30	456	Tipo4	0094	195	Tipo2	0302	175
31	465	Tipo4	0334	203	Tipo1	0081	184
32	477	Tipo5	0015	189	Tipo2	0252	183
33	487	Tipo5	0365	194	Tipo1	0061	186
34	496	Tipo5	0355	199	Tipo2	0252	189
35	505	Tipo5	0105	201	Tipo1	0091	192
36	514	Tipo5	0325	209	Tipo1	0061	194
37	520	Tipo5	0215	210	Tipo3	0043	198
38	532	Tipo5	0135	212	Tipo1	0171	202
39	544	Tipo5	0195	217	Tipo1	0061	209
40	556	Tipo5	0275	223	Tipo3	0043	210
41	567	Tipo1	0191	216	Tipo4	0174	213
42	580	Tipo3	0183	219	Tipo1	0161	218
43	593	Tipo3	0043	223	Tipo1	0341	222
41	567	Tipo1	0191	216	Tipo4	0174	213
42	580	Tipo3	0183	219	Tipo1	0161	218
43	593	Tipo3	0043	223	Tipo1	0341	222
^Tiempo 3:42:57							

Tabla 6.4 Intercambios – simulación4.

6.4 Codificación del Algoritmo

Una de las propiedades que ofrece el formalismo de las Redes de Petri como herramienta de modelado es la capacidad de describir modularmente los subsistemas.

La codificación se realiza con el diseño de la programación Bottom-up las partes individuales se diseñan con detalle y luego se enlazan para formar componentes más grandes, que a su vez se enlazan hasta que se forma el sistema completo.

Inicio

Asignación Aleatorio
Asignación de TiempoX
Anexo de Automóviles Anteriores
ProcesoY_Colas

Mientras EstatusTiempo<>stock vacío

Procedimiento Identificación
Procedimiento Intercambio
Termina Mientras

FIN

El primer procedimiento realiza una asignación aleatoria de números enteros a todos los vehículos que se encuentran en el stock inicial, con el objetivo de establecer una secuencia inicial.

Procedimiento Asignación Aleatoria

Variables:

j=1...n tipos de automóviles
 i=1,...mj automóviles del tipo j
 Mj: Max(mj)
 s=1,...hs : secuencia 1
 SubClnicial: entero

Inicio

Lee información de número de automóviles en el subconjunto
 Para j=1 hasta n
 Para i=1 hasta Mj
 Posición(i,j)=aleatorio
 Fin Para
 Fin Para

Ordena de acuerdo a aleatorio Posición(i,j):
 Asigna secuencia hs
 Fin

Una vez que se ha determinado la secuencia inicial en la cual entrarán todos los vehículos al proceso, es necesario establecer los tiempos de entrada y salida para cada uno de los vehículos de la estación de trabajo donde se lleva a cabo el primer proceso.

Procedimiento Asignación de TiempoX

Variables:

TiempoPX, SubCInicial, IPX(s), FPX(s),
 s=1,...hs : secuencia 1

Inicio

Lee información secuencia 1
 Para s=1, IPX(s)=1, FPX(s)=TiempoPX
 Para s>1 hasta SubCInicial
 IPX(s)=FPX(s-1)+1+TiempoPX
 FPX(s)=IPX(s)+TiempoPX

Fin Para
 Ordena por posición(i,j)

Fin

En este punto se revisa la secuencia de los automóviles que han terminado el proceso X, el día anterior, y se establece una nueva secuencia de entrada a la siguiente estación de trabajo.

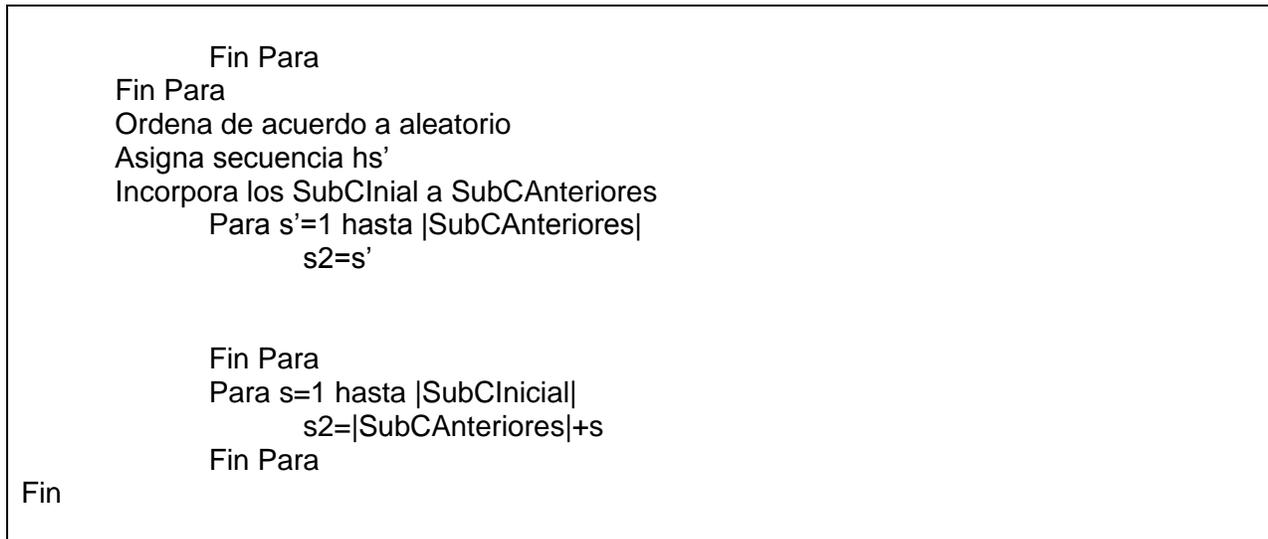
Procedimiento Anexo de Automóviles Anteriores

Variables:

SubCInicial
 SubCAnteriores
 ja=1...na tipos de automóviles
 ia=1,...maj automóviles del tipo j
 Maj: Max(maj)
 s=1,...hs: secuencia 1 (inicial)
 s'=1,...hs' : secuencia de automóviles anteriores
 s2=1,...h2s secuencia 2 (incluyendo los anteriores)

Inicio

Lee información de número de automóviles anteriores
 Para ja=1 hasta na
 Para ia=1 hasta Maj
 Posición(i,j)=aleatorio



Una vez establecida la secuencia de entrada, se registran los tiempos para el siguiente proceso, en este momento la producción se divide en las diferentes líneas a producir de vehículos, Tipo1, Tipo2, Tipo3 y Tipo2, en cada una de ellas se establece una secuencia propia basada en la herencia de las secuencias anteriores.

En este procedimiento se establecen el estatus de colas en el tiempo t para cada Tipo de vehículo de acuerdo con los tiempos de fin de proceso X. Así mismo se registra el estatus de la estación de trabajo.

Procedimiento ProcesoY_Colas

Variables:

j=1...n tipos de automóviles
 i=1,...mj+maj automóviles del tipo j
 SubClnicial
 SubCAnteriores
 s2=1,...h2s secuencia 2 (incluyendo los anteriores)
 's(j)=1,...h's(j) secuencia por Tipo
 TiempoPX, IPX(s2),FPX(s2),
 TiempoPYj, IPY['s(j)],FPY['s(j)]
 Tiempo
 's(j).contador, Cola[Tiempo(j)]: Entero
 Tiempo.pieza, EstatusCola, EstatusTiempo: Texto

Inicio

Agrupa por tipo para establecer nueva secuencia por tipo: h's(j)
Revisa de tiempo en proceso X
 Para s2=1 hasta |SubCAnteriores|

```

    IPX(s2)=0, FPX(s2)=0
Fin Para
Para s2=1+|SubCAnteriores| hasta |SubCInicial|
    Lee los tiempos de procesos establecidos en Asignación de TiempoX
Fin Para

Para j=1 hasta n
    Para 's(j)=1, IPY['s(j)]=1, FPY['s(j)]=TiempoPYj
    Para 's(j)>1

        Si IPX['s(j)]=0 entonces IPY['s(j)]=FPY['s(j)-1]+1 y
            FPY['s(j)]=IPY['s(j)']+TiempoPYj
        Fin Si
        Si IPX['s(j)]<>0 entonces IPY['s(j)]=max(FPX['s(j)],FPY['s(j)-1]+1) y
            FPY['s(j)]=IPY['s(j)']+TiempoPYj
        Fin Si
    Fin Para
Fin Para

Establece reloj y evalua
    Si IPY['s(j)] <=Tiempo<=FPY['s(j)]
        Tiempo.pieza=posición(i,j)
        EstatusTiempo=En proceso
    Fin Si
    Si IPY['s(j)] >Tiempo>FPY['s(j)]
        Tiempo.pieza=0
        EstatusTiempo=""
    Fin Si

    Si IPY['s(j)+1]< FPY['s(j)] entonces
        EstatusCola=Pieza en cola
    Fin Si
    Si IPY['s(j)+1] > FPY['s(j)] entonces
        EstatusCola=Linea vacía
    Fin Si

Evalua cada posición en el tiempo
    Para Tiempo=1 hasta max[FPY['s(j)] ]
    Si 's(j) es tal que IPY['s(j)] <=Tiempo<=FPY['s(j)] entonces 's(j).contador=1
        Cola[Tiempo(j)]=suma ('s(j).contador)
    Fin Si

Evalua todas las colas
    En Tiempo T:
    Si posición(i,j) <>0 y Tiempo<=SubCAnteriores*TiempoPX
        entonces EstatusTiempo=TG

```

```

Fin Si
Si posición(i,j)<>0 y Tiempo>SubCAnteriores*TiempoPX
    entonces EstatusTiempo=ocupado
Fin Si
Si posición(i,j)=0 para alguna j entonces EstatusTiempo=cero
Fin

```

El procedimiento identifica por un lado la estación de trabajo que ha caído a cero, y por otro, la línea de producción tipo, que tardará más en consumirse, de acuerdo al número de elementos que haya en la línea de espera y el tiempo en que tarden éstos elementos en ser procesados.

Procedimiento Identificación

Variables:

```

j=1...n tipos de automóviles
i=1,...mj+maj automóviles del tipo j
s2=1,...h2s secuencia 2 (incluyendo los anteriores)
's(j)=1,...h's(j) secuencia por Tipo
TiempoPYj, IPY['s(j)],FPY['s(j)]
Tiempo.caida: Tiempo en que se registra cero en EstatusTiempo
j*,i*: posición de la pieza que provoca la caída
s2* secuencia de la posición i*,j* en el conjunto total
's(j)* secuencia de la posición i*,j* del tipo j
jc,ic: posición de la pieza de intercambio
s2c secuencia de la posición ic,jc en el conjunto total
's(j)c secuencia de la posición ic,jc del tipo j
LineaTardada(j): Tipo de pieza cuya cola tardará mas tiempo en pasar al proceso Yj
Cola[Tiempo(j)]
Tiempo:Entero
Tiempo.pieza, EstatusCola, EstatusTiempo: Texto

```

Inicio

```

Si EstatusTiempo=cero entonces
    Identifica Tipo j, Tiempo, secuencia s2, 's(j) y posición
    s2= s2*, 's(j)='s*(j), posición(i,j)=posición(i*,j*)
    Tiempo=TiempoCaida
Fin Si
LineaTardada(j)=Cola[Tiempo(j)]*TiempoPYj
jc=Max(LineaTardada(j))
Identifica Tipo jc secuencia s2, 's(j) y posición
    s2= s2c, 's(j)='s(j)c posición(i,j)=posición(ic,jc);

```

Fin

Cuando se han identificado los elementos, se procede al intercambio, se permutan las secuencias y se ajustan en las líneas de producción cuyos elementos fueron modificados, los tiempos de inicio del proceso especializado son adaptados a la nueva secuencia, y se registran como consumidos aquellos elementos cuyo tiempo final del proceso especializado. Aquellos elementos cuyos registros de tiempo son menores a los tiempos establecidos en las unidades de intercambio son marcados como consumidos.

Procedimiento Intercambio

VARIABLES:

$j=1 \dots n$ tipos de automóviles
 $i=1, \dots, m_j + m_{aj}$ automóviles del tipo j
 $s_2=1, \dots, h_2s$ secuencia 2 (incluyendo los anteriores)
 $'s(j)=1, \dots, h's(j)$ secuencia por Tipo
 TiempoPX, IPX(s_2), FPX(s_2),
 TiempoPY $_j$, IPY[' $s(j)$ '], FPY[' $s(j)$ ']
 Tiempo.caida: Tiempo en que se registra cero en EstatusTiempo
 s_2^* secuencia de la posición i^*, j^* en el conjunto total
 $'s(j)^*$ secuencia de la posición i^*, j^* del tipo j
 s_2^c secuencia de la posición i^c, j^c en el conjunto total
 $'s(j)^c$ secuencia de la posición i^c, j^c del tipo j
 Cola[Tiempo(j)]
 Tiempo:Entero
 Tiempo.pieza, EstatusCola, EstatusTiempo: Texto

Inicio

Intercambia posiciones en secuencia
 $s_2^* = s_2^c$
 $s^c = s_2^*$
Establece Tiempo de proceso de con el intercambio

Si TiempoCaida < Tiempo entonces Posicion(i, j) = Consumido

Para $j=1$ hasta n

Para ' $s(j)$ ' asociado a la posición(i, j) tal que $s_2 = s_2^c$,
 $IPX[s_2] = \text{TiempoCaida} - 1 - \text{TiempoX}$
 $FPX[s_2] = IPX[s_2] + \text{TiempoPX}$
 $IPY['s(j)'] = \text{TiempoCaida}$, $FPY['s(j)'] = \text{TiempoCaida} + \text{TiempoPY}_j$

Fin Para

Para ' $s(j)$ ' asociado a la posición(i, j) tal que $s_2 > s_2^c$

$IPX(s_2) = FPX(s_2^c - 1) + 1$
 $FPX(s_2) = IPX(2s) + \text{TiempoPX}$

$IPY = FPY['s(j)'] - 1 + 1$
 $IPY = \max(FPX['s(j)'], FPY['s(j)'] - 1) + 1$
 $FPY['s(j)'] = IPY['s(j)'] + 1 + \text{TiempoPY}_j$

Fin Para

Fin Para

Evalua cada posición en el tiempo

Para Tiempo=TiempoCaida hasta $\max(\text{FPY}[\text{'s}(j)])$

Si $\text{'s}(j)$ es tal que $\text{IPY}[\text{'s}(j)] \leq \text{Tiempo} \leq \text{FPY}[\text{'s}(j)]$ entonces $\text{'s}(j).\text{contador}=1$

Cola[Tiempo(j)]=suma ($\text{'s}(j).\text{contador}$)

Fin Si

Evalua todas las colas

Si posición(i,j) \neq (C,0) y $\text{Tiempo} > \text{TiempoCaida}$, y $\text{IPY}[\text{'s}(j)] \leq \text{Tiempo} \leq \text{FPY}[\text{'s}(j)]$

entonces EstatusTiempo=ocupado

Fin Si

Si posición(i,j)=0, posición(i,j) \neq (C) y $\text{Tiempo} > \text{TiempoCaida}$ para alguna j

entonces EstatusTiempo=cero

Fin Si

Si posición(i,j)=0, Cola[Tiempo(j)]=0 para toda j

entonces EstatusTiempo=stock vacío

Fin Si

Fin

Capítulo 7

CONCLUSIONES

La representación del problema de secuenciación en la industria de automóviles a través del formalismo de las Redes de Petri Coloreadas es una nueva forma de abordar el problema. Las Redes de Petri han probado ser una herramienta exitosa en el modelado de sistemas logísticos de producción complejos representando formalmente paralelismo y sincronización, además es posible efectuar un análisis cuantitativo del sistema, estudiando aspectos estructurales tales como situaciones de bloqueo.

Sin embargo, debido al elevado número de posibles configuraciones en un sistema basado en secuencias, el árbol de cobertura representado por la RPC, es muy grande, dificultando la apertura de todos los estados posibles.

El algoritmo presentado establece restricciones que acotan el árbol cuando un nodo hoja de solución no factible es alcanzado, de tal forma que el número de permutaciones se reduce, así mismo utiliza mecanismos (intercambio de elementos), para que alcanzado un nodo hoja de solución no factible, pueda reactivarse el sistema mediante la permutación de piezas que no han salido del mismo, de esta forma la heurística no pierde soluciones factibles, al intercambiar los elementos, se genera un estado que habría sido alcanzado con una configuración inicial diferente.

En forma particular, se busca solucionar el problema de secuenciación causado por la inactividad de alguna de las células de trabajo. La inactividad se produce cuando no han llegado piezas (vehículos) a la estación de trabajo, lo cual provoca el desequilibrio del sistema. El método plantea un procedimiento de intercambio de piezas dentro de la secuencia como solución inmediata a la presentación del problema. De esta forma la toma de decisiones de configuración de entrada de los diferentes tipos de vehículos es apoyada por la reactivación del intercambio, que provee la continuidad en momentos de ruptura de stock.

Es importante mencionar que el análisis de la Red de Petri Coloreada del problema de secuenciación, permite identificar gracias a los atributos de cada pieza / vehículo, aquella que debido a su lugar en la secuencia inicial ha provocado la interrupción del sistema.

La corrección de la secuencia en el momento de la ruptura de stock representa una mejora en la planificación de la producción, al tener todas las estaciones de trabajo produciendo.

El algoritmo propuesto, realiza los intercambios entre aquellos elementos que se encuentran en un conjunto factible, este conjunto está formado por las piezas cuyo inicio en el proceso especializado es mayor al tiempo de detección de la caída a cero. Por lo tanto conforme el reloj avanza, este conjunto se vuelve más pequeño y como consecuencia el número de posibles intercambios. Lo cual marca una diferencia significativa al compararlo con el número de permutaciones de la

secuencia completa al inicio. Otra ventaja de este método es la capacidad de identificar la línea de espera más grande de acuerdo al tiempo de proceso en cada estación de trabajo especializada.

Futuras líneas de investigación

El problema de secuencia, se encuentra en diferentes campos, con lo cual la aplicación del algoritmo a otras áreas es factible.

El modelo plantea sólo un punto de decisión donde los elementos / entidades pueden intercambiarse, al tener un comportamiento igual en el proceso común; como línea de trabajo se propone:

- El estudio de un método de reactivación la a partir de dos puntos de decisión, así como el análisis de la complejidad del algoritmo ante n puntos de decisión.
- La variante del modelo de un sistema, con tiempos diferentes en el proceso común.
- Desde el punto de vista matemático, la evaluación de la convergencia del algoritmo.
- Desde el punto de vista experimental, el desarrollo de nuevas heurísticas para determinar alternativas de intercambio.

BIBLIOGRAFIA

AARTS, E. y LENSTRA, J. (1993). Local Search in Combinatorial Optimization. John Wiley & Sons.

ACEVEDO.J.A; GOMEZ, M (2001). Logística moderna y la competitividad empresarial. Cuba. 5, 6,18 p.

ADEDEJI, B .(2006). Handbook of industrial and systems engineering, Volumen 56. Taylor & Francis Group.

BALLOU, R. H. (1991). Logística Empresarial. Control y Planificación Madrid., Ediciones Díaz de Santos.

BARBER, F., SALIDO, MA (2003) Introducción a la Programación de Restricciones.; Inteligencia Artificial: Revista Iberoamericana de Inteligencia Artificial. Nº 20, 2003.

CHENG, J., LU, Y., PUSKORIUS, G., BERGEON, S., XIAO, J., (1999) Vehicle sequencing based on evolutionary computation Evolutionary Computation (CEC 99), vol. 2, IEEE, pp. 1207–1214.

DAVIS MARK, FUNDAMENTOS DE DIRECCION DE OPERACIONES.MCGRAW HILL.

DORIGO, M., STNTZLE,T., (2005) Ant Colony Optimization, MIT Press.

Dpto. Estadística, Econometría, Inv. Op. y Org. de Empresas. El Sistema Productivo de la Empresa: Planificación, Programación y Control de la Producción UNIVERSIDAD DE CÓRDOBA ESCUELA POLITÉCNICA SUPERIOR. <http://www.uco.es/~p42abluj/web/indice.htm>.

DREXL, A., KIMMS, A. (2001). Sequencing jit mixed-model assembly lines under station load and part-usage constraints, Management Science 47 (3) 480–491.

DREXL, A., KIMMS, A., MATTHIESSEN, L. (2006). Algorithms for the car sequencing and the level scheduling problem, Journal of Scheduling 9 (2).

EVEREST, E. A., RONALD J. EBERT (1991). Administración de la producción y operaciones. México., Ed. Prentice Hall. Hispanoamérica S.A.

ELA European Certification Board for Logistics (2004) . <http://www.elalog.org/>.

GARCÍA SÁNCHEZ, L. (2004)Trabajo de Diploma “Investigación exploratoria de la gestión de aprovisionamiento”.

GENT, I.P., (1998). Two results on car-sequencing problems, Technical report APES-02-1998 <<http://www.dcs.st-and.ac.uk/apes/reports/apes-02-1998.ps.gz>>.

- GOTTLIEB, J., PUCHTA, M., SOLNON, C. (2003). A study of greedy, local search and ant colony optimization approaches for car sequencing problems, in: Applications of Evolutionary Computing LNCS, vol. 2611, Springer, , pp. 246–257.
- GRAVEL, M, GAGNE´, C., PRICE, W.L. (2005). Review and comparison of three methods for the solution of the car-sequencing problem, Journal of the Operational Research Society 56 (11) 1287–1295.
- GUASCH, A., PIERA, MA; CANOVAS J.; FIGUERAS, J. (2005). Modelado y Simulación: Aplicación a Procesos Logísticos de Fabricación y Servicios. Alfaomega grupo Editor. México.
- HALEVI, G. (2003). Process and operation planning. Kluwer academic Publishers.
- HEIZER, J. y RENDER, B. (2001). Dirección de la Producción. Decisiones Tácticas. Prentice-Hall. Pearson Education. Madrid.
- HINDI K.S., PLOSZAJSKI, M.G. (1994) Formulation and solution of a selection and sequencing problem in car manufacture, Computers and Industrial Engineering 26 (1) 203–211.
- INGNIZIO, J. y CAVALIER. (1994), Linear Programming. Prentice Hall.
- JENSEN K. (1997). Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Vol. 1,2,3. Springer-Verlag, Berlin.
- KIS,T. (2004). On the complexity of the car sequencing problem, Operations Research Letters 32 331–335.
- KRISTENSEN, L.M., CHRISTENSEN, S. (2004). Implementing Coloured Petri Nets Using a Functional Programming Language. Higher-Order and Symbolic Computation, Vol. 17. Kluwer Academic Publishers, The Netherlands.
- LEE,J.H.M., LEUNG,H.F. WON, H.W. (1998) Performance of a comprehensive and efficient constraint library using local search, in: 11th Australian JCAI, LNAI, Springer-Verlag.
- MICHEL, L., VAN HENTENRYCK, P. (2002). A constraint-based architecture for local search, in: OOPSLA'02: Proceedings of the 17th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, ACM Press, New York, NY, USA, , pp. 83–100.
- MONKS JOSEPH, ADMINISTRACION DE OPERACIONES. MCGRAW HILL.
- MORAGA, WHITEHOUSE y DEPUY. (2003) META-RAPS: un enfoque de solución eficaz para problemas combinatorios. Revista Ingeniería Industrial - Año 2, N°1 - Segundo Semestre 2003.

- NARCISO (2007), M. Metodología para la Resolución de problemas de optimización mediante la exploración del espacio de estados generado a partir de modelos de redes de petri coloreadas. Tesis Doctoral.
- NEVEU,B., TROMBETTONI, G., GLOVER, F.(2004). Id walk: A candidate list strategy with a simple diversification device, in: Proceedings of CP'2004LNCS, vol. 3258, Springer-Verlag, pp. 423–437.
- PARELLO B.D., KABAT W.C., WOS, L. (1986) Job-shop scheduling using automated reasoning: A case study of the car sequencing problem, *Journal of Automated Reasoning* 2 1–42.
- PERRON, L., SHAW, P., (2004) Combining forces to solve the car sequencing problem, in: Proceedings of CP-AIOR' 2004LNCS, vol. 3011, Springer, pp. 225–239.
- PERRON, L., SHAW, P., FURNON, V., (2004). Propagation guided large neighborhood search, in: Proceedings of Principles and Practice of Constraint Programming (CP'2004)LNCS, vol. 3258, Springer, pp. 468–481.
- PERRON,L. SHAW, P. (2004). Combining forces to solve the car sequencing problem, in: Proceedings of CP-AIOR' 2004LNCS, vol. 3011, Springer, pp. 225–239.
- PERRON,L., SHAW, P., FURNON,V. (2004). Propagation guided large neighborhood search, in: Proceedings of Principles and Practice of Constraint Programming (CP'2004) LNCS, vol. 3258, Springer, 2004, pp. 468–481.
- PIERA MA, GUASCH A. (2003) Model based decisions approach reactive modeling to improve logistic systems.
- PROTH, JM; XIE, X (1996) Petri Nets: A Tool for Design Management of Manufacturing Systems. Ed John Wiley & Sons, England.
- PUCHTA, M., GOTTLIEB, J., (2002) Solving car sequencing problems by local optimization, in: EvoWorkshops, LNCS, vol. 2056, Springer-Verlag, , pp. 132–142.
- REGIN, J.-C. PUGET,J.-F. (1997). A filtering algorithm for global sequencing constraints, in: CP97LNCS, vol. 1330, Springer-Verlag, 1997, pp. 32–46.
- REINELT, G. (1994). The Traveling Salesman: Computational Solutions for TSP Applications. Lectures. Notes in Computer Science, Springer-Verlag.
- SABLÓN, N. QUINTANA, L., HERNÁNDEZ V., MEDEROS, R., CASTELLANOS, J.A. (2009). La cadena de suministro elemento clave de la gestión logística. Canales de distribución y administración logística.

SILVA M., VALETTE R. (1989). Petri nets and flexible manufacturing. Lecture notes in computer science, Vol. 424. Advances in Petri Nets.

SMITH, B. (1996). Succeed-first or fail-first: A case study in variable and value ordering heuristics, in: Third Conference on the Practical Applications of Constraint Technology PACT'97, pp. 321–330.

SOLNON, C., (2000). Solving permutation constraint satisfaction problems with artificial ants, in: Proceedings of ECAI'2000, IOS Press, Amsterdam, The Netherlands, 2000, pp. 118–122.

SOLNON, C., CUNG, V-D., NGUYEN A., ARTIGUES, C.,. (2008). The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. European Journal of Operational Research 191 912–927.

TORRES, M. (2004) Logística. Temas seleccionados. Cuba, Feijoo

TSANG, E.P.K. (1993). Foundations of Constraint Satisfaction, Academic Press, London, UK.

VAN HOEVE, W.-J., PESANT,G. ROUSSEAU, L.-M., SABHARWAL, A. (2006) Revisiting the sequence constraint, in: 12th International Conference on Principles and Practice of Constraint Programming (CP'2006) pp. 620–634.

WARWICK, T., TSANG, E., (1995) Tackling car sequencing problems using a genetic algorithm, Journal of Evolutionary Computation– MIT Press 3 (3) 267–298.

WHITEHAUSE, G. y DE PUY, G. (2001), Solving Constrained Multiple Resource Networks Both Forward and Backward Using Brooks Algorithm, Project Management Journal, Vol. 32, No. 4, pp. 24-31.

YU, G. (1998). Industrial Applications of Combinatorial Optimization. Kluwer Academic Publishers.

ZIMMERMAN AK; DALKOWSKI, HOMMEL (1996). A case of study in modeling and performance evaluations of manufacturing systems using colored Petri nets, in proceedings of the 8th European Simulation Symposium (ESS'96).