Red de Petri simulada en NetLogo

Karen Hernández Rueda
Universidad de Guadalajara, Zapopan, Jalisco, México, karenhr@cucsur.udg.mx

María Elena Meda Campaña
Consejero de la Facultad
Universidad de Guadalajara, Zapopan, Jalisco, México, emeda@cucea.udg.mx

RESUMEN

Tanto las redes de Petri como los algoritmos genéticos son herramientas que proporcionan ventajas en el modelado y simulación de diversos sistemas por sus características implícitas. Aunque ambas han incrementado su uso en diversas investigaciones y se han complementado con el fin de mejorar resultados, todavía no se cuenta con una herramienta de red de Petri en Netlogo para trabajar con problemas generalizados. Por lo que este trabajo presenta una red de Petri modelada y simulada con NetLogo (algoritmo genético) así como las ventajas y desventajas de la implementación, con el objetivo de que sirva de base para generalizar el diseño de redes de Petri con algoritmos genéticos para cualquier tipo de sistema y para su respectivo análisis basado en una función objetivo. Asimismo se presenta una propuesta para generalizar el modelado de las redes de Petri usando algoritmos genéticos.

Palabras claves: NetLogo, red de petri, algoritmo genético, simulación

ABSTRACT

Both Petri nets and genetic algorithms are tools that provide advantages in modeling and simulation of various systems by implicit features. Although both have increased their use in various researches, and have been combined in order to improve results, have not been used yet implemented through one another to work with widespread problems. Therefore, this paper presents a Petri net modeled with NetLogo (genetic algorithm) pointing out the advantages and disadvantages of the implementation with the aim of serving as the basis for the design of generalized Petri nets with genetic algorithms to any system and their analysis based an objective function. It also presents a proposal for modeling generalized Petri nets using genetic algorithms.

Keywords: NetLogo, petri net, genetic algorithm, simulation

1. INTRODUCCIÓN

Las redes de Petri (RP) se pueden representar como gráficos que pueden simbolizar procesos paralelos y asíncronos. Muchas ocasiones lo que se busca con estas redes es optimizar un sistema, que incluye los componentes y sus valores de ajuste. Cuando se modelan redes de gran tamaño, parece que lo más eficaz es conseguir la fuerza computacional de los algoritmos evolutivos con el fin de buscar la configuración de la red óptima por medio de algoritmos genéticos (AG). Los algoritmos genéticos se han utilizado, por ejemplo, para resolver un problema de accesibilidad y programación de tiempo en RP. Sin embargo, no se han utilizado para optimizar las ventajas de diseño. Una de los motivos de este trabajo es buscar la forma de realizar esta tarea en un futuro próximo. Por un lado, las RP son un formalismo que permiten modelar complejos numerosos problemas del mundo reales (tales como sistemas de tráfico y circuitos electrónicos), donde una de sus potencialidades es la construcción del mecanismo de concurrencias, que permiten una descripción formal de los sistemas que contienen procesos paralelos. Y por otra parte, los AG son un formalismo que permiten modelar problemas típicos de

optimización combinatoria tales como asignación, transporte, pronósticos, entre otros y que permiten explorar un espacio de soluciones sin usar tanta memoria de procesamiento.

Este trabajado ha sido inspirado por los trabajos que se han realizado aplicando tanto el modelado de las redes de Petri como los algoritmos genéticos en infinidad de problemas como en (O'Neill et al, 2012), en (Mayo and Beretta, 2010), en (Lihong et al, 2009), en (Chien and Chen, 2007), en (Caballero and Gonzalo, 2006) y en (Reid 1998). El objetivo final es contar con un algoritmo generalizado para implementar una RP y analizar diversos problemas, con la modificación de una función objetivo (FO) deseada. Como primera etapa, lo que se busca es revisar las ventajas y las desventajas de unir ambas herramientas, a través de la implementación de una RP con AG. Aquí se presenta la forma de modelar una RP usando la herramienta de NetLogo como algoritmo genético, sin establecer una FO en particular. Sin embargo, se presenta una propuesta de cómo podría generalizarse el modelo de la RP para analizar otros tipos de sistemas.

Primeramente se revisan los trabajos en los que se han usado ambos formalismo, luego se presentan las bases de modelado de RP como de los algoritmos genéticos, posteriormente se presenta la RP modelada en NetLogo con el algoritmo que se implementó y luego se muestran los resultados de la simulación. Por último, se presentan conclusiones y trabajo futuro con las ventajas y desventajas de la simulación, y la propuesta para modelar cualquier red.

2. ANTECEDENTES

Existe mucha literatura en las que se analizan diversos problemas con las RP y los AG como en (Mayo and Beretta, 2010), en (Wolters and Steffens, 2008), en (Chein and Chen, 2007), en (Lihong et al, 2009) o en (Mauch, 2003). A continuación se describen brevemente alguno de los trabajos que involucran ambas herramientas.

En (Mayo and Beretta, 2010) se utilizan tanto las redes de Petri como algoritmos genéticos para proponer un método que construye automáticamente modelos de redes de Petri de interacciones de genes no lineales. Por una parte, se configura una red de Petri parcial inicial con repetidas sub-redes que modelan genes individuales y un conjunto de restricciones. Estas restricciones caracterizan a la clase de red de Petri que se desean. Después, esa estructura y sus restricciones se usan como entrada de un algoritmo genético. El algoritmo genético busca una arquitectura de RP que es tanto un superconjunto de la red inicial como también todas las restricciones dadas.

En (Wolters and Steffens, 2008) se presenta un enfoque de aprendizaje máquina para la adquisición automática de agentes de comportamientos en simulación basada en agentes. Las RP se usan para especificar los niveles de abstracción, y en este caso, se usan para especificar el comportamiento de un agente así como para implementarlo. Considera que las redes de Petri son un lenguaje de representación prometedor para aprender comportamiento de agentes y el enfoque que se presenta, aplica un algoritmo genético en las estructuras gráficas que están representados por un extensión de las redes de Petri, llamado "Lenguaje para Modelado de Procesos y Escenarios basado en Agentes" (LAMPS.) Un fragmento de un LAMPS usa cuatro conceptos: el agente *Informante A* observa el lugar del *Enemigo* marcado. Si el lugar contiene una marca, el agente ejecuta una acción de *Combate*, que envía una marca al lugar de la *Resistencia* rota. La tarea del algoritmo de aprendizaje (basado en AG) es reunir bloques de construcción LAMPS (transacciones y lugares), en un orden apropiado.

En (Marwan et al, 2007) se modifica el concepto de RP para definir formalmente una estructura reguladora, que propiamente modela la topología de la red y el comportamiento dinámico del sistema interés. Ya que, aunque las RP la utilizan la comunidad de biología para representar las redes de interacción porque permiten combinar los parámetros y procesos en los diferentes niveles de complejidad dentro de un modelo único y coherente, tienen dificultades para predecir la dinámica del sistema.

En (Caballero and Gonzalo, 2006) se propone un enfoque para modelar sistemas de manufactura flexible y generar programas de producción activos orientados a la minimización de la tardanza ponderada de los trabajos, utilizando las RP y AG. Modelan el sistema de manufactura flexible y posteriormente, usan los AG para solucionar los problemas de programación. Proponen un algoritmo REPAG nombrado así por la conjunción de RP y AG. Lo que se hace es implementar un AG con la estructura de la RP asociada con los sistemas de manufactura flexible que cumplen todas las restricciones de procedencia de los trabajos y de capacidad de las

máquinas. Allí el AG codifica una secuencia de disparos de transiciones que transforma el marcado inicial hasta alcanzar el marcador final deseado, a través de un cromosoma. Este cromosoma permite generar todo el espacio posible de soluciones, garantiza un solución factible después de ser recombinado o mutado y no necesita de estrategias de reparación. En este caso la función aptitud representa la tardanza ponderada de los trabajos.

La programación de bucles en paralelo se ha representado con redes de Petri usando algoritmos genéticos, en (O'Neill et al, 2012), se introduce una aplicación de AG para operaciones repetitivas (como en manufactura, control de operación), y se usa una representación de RP de la operación repetitiva que habilita un AG que busca espacios de la operación repetitiva que sean válidos para incorporar fuentes y limitaciones de dependencia. Cada genoma representa el espacio de las asignaciones de retardo posibles a los arcos de la red de Petri. La función de costo simula la RP para identificar el programa iterativo, el cual es entonces evaluado.

(Prashant et al 2001) usa una red de Petri como modelado y herramienta de "sheduling" en administración de proyectos donde se propone una extensión de las RP para las actividades de "scheduling" en una decisión y el uso de una matriz para el movimientos de los marcados. Un AG es usado para encontrar la mejor solución, basado en búsqueda heurística, considerando la forma de lidiar con los recursos limitados en la gestión de proyectos. El AG se usa como una técnica de búsqueda rápida dirigida a encontrar una solución mejorada en un espacio de búsqueda compleja en el caso de las redes de decisión. Muestra que un software de redes de Petri junto con una búsqueda basada en un AG puede hacer frente a proyectos que tengan actividades de decisión con múltiples modos, las limitaciones de múltiples recursos y puede administrar eficientemente la preferencia de las actividades, se puede planear, programar y asignar recursos en los proyectos.

El trabajo de (Mauch, 2003) examina las redes de Petri como una clase de red de bajo nivel de redes de transición de lugar, como estructuras que se someten a la evolución. Aquí se introduce el enfoque de la evolución de las RP conocido como PNE. Como las RP se pueden considerar estructuralmente, grafos bipartitos especializados consideran a las PNE una forma de programación genética.

Sin embargo, estos problemas resuelven un problema específico y no es claro cómo se puede modelar las RP con AG para dar solución a diversos problemas, por lo que el objetivo de este trabajo es mostrar cómo realizar el modelado usando NetLogo como una herramienta de AG. Antes de mostrar la implementación, primero se presentan las definiciones generales tanto de RP como de AG.

3. FUNDAMENTOS

A continuación se presentan los conceptos básicos de las redes de Petri y los algoritmos genéticos.

3.1 REDES DE PETRI

Las RP como herramientas gráficas y matemáticas, proporcionan un entorno uniforme para el modelado, análisis formal y el diseño de sistemas de eventos discretos (Silva, 1985). Una de las ventajas principales de usar modelos de RP es que el mismo modelo se puede utilizar para el análisis de las propiedades de comportamiento y evaluación del desempeño. Las RP son una herramienta de modelado gráfico y matemático aplicable a muchos sistemas. Como una herramienta gráfica, las RP se pueden utilizar como una ayuda visual de comunicación similar a los diagramas de flujo, diagramas de bloques, y redes, y proporcionar un medio de comunicación entre el usuario; por lo general, los requisitos del ingeniero y el cliente. Como una herramienta matemática, es posible establecer ecuaciones de estado, ecuaciones algebraicas, y otros modelos matemáticos que rigen el comportamiento de los sistemas.

Una RP consiste de una estructura de red (un dígrafo bipartito), una descripción de estado (el marcado) y una regla de transición (el juego de marcas). Un ejemplo de una RP se muestra en la siguiente figura 1:

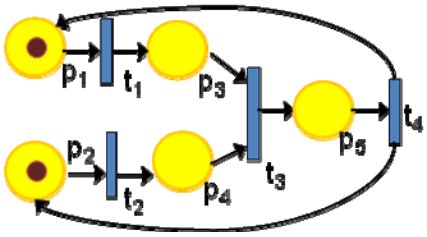


Figura 1: Ejemplo de una red de Petri

La red se representa por dos clases de vértices: círculos que representan lugares (p1,p2,p3,p4), a los que se les asocian acciones o salidas del sistema que se desea modelar y barras o rectángulos (t1,t2,t3,t4) que representan transiciones, a los que se les asocian eventos y acciones o salidas. Los lugares a su vez pueden estar marcados, es decir, se distribuyen marcas (puntos) dentro de los lugares. Un marcado inicial sería una distribución inicial de marcas. Los lugares de entrada/salida son lugares cuyos arcos llevan a (salen de) una transición t_j y se consideran de entrada a (salida de) t_j. La evolución de marcas consiste en una simple regla de dos partes:

- transición habilitada: los lugares de entrada deben al menos tener tantas marcas como el peso de los arcos de entrada
- disparo de transición: elimina tantas marcas como el peso del arco de los lugares de entrada a una transición habilitada y agrega tantas marcas a los lugares de salida como el peso del arco de salida lo indica.

En general, ${}^\bullet t_j$ representa el conjunto de lugares p_i, tal que $I(p_i,t_j)\neq 0$ y $t_j^\bullet$ el conjunto de lugares tal que $O(p_i,t_j)\neq 0$. Análogamente, ${}^\bullet p_i$ representa el conjunto de transiciones t_j tal que $O(p_i,t_j)\neq 0$ y $p_i^\bullet$ representa el conjunto de transiciones t_j tal que $I(p_i,t_j)\neq 0$. Sea X (Y) un subconjunto de lugares de P (transiciones de T), entonces ${}^\bullet X$ y $X^\bullet$ (${}^\bullet Y$ y $Y^\bullet$) denotan el conjunto de lugares p_i (transiciones t_j) tal que $I(p_i,t_j)\neq 0$ y $O(p_i,t_j)\neq 0$, respectivamente, para todo $t_j \in X$ ($p_i \in Y$). La matriz de incidencia de G es $C=[c_{ij}]$, donde $c_{ij}= O(p_i,t_j) - I(p_i,t_j)$. La función de marcado $M: P \rightarrow Z^+$ es un mapeo desde cada lugar a un entero no negativo representado por el numero de marcas (representado en forma de puntos) dentro de cada lugar. El marcado de una red de Petri PN es generalmente expresado como un vector de entrada n.

3.2 ALGORITMOS GENÉTICOS

Los AG son técnicas de búsqueda estocásticas basado en los mecanismos de evolución natural (Goldberg, 1989). Los AG usan una población existente (conjunto de individuos que pueden representan diferentes cosas) sobre la cual se aplican operadores y estrategias con el objetivo de encontrar una nueva población que sea mucho mejor que la población existente. La idea es que sólo las mejores poblaciones se enriquezcan (evolucionen) para que sobrevivan y se preserven las mejores características (función de aptitud u objetivo) de los individuos de las anteriores generaciones.

Un AG es una técnica iterativa basado en una población, que usa una estrategia de selección y operadores, conocido como mutación y recombinación, para generar y explorar nuevos puntos en el espacio de soluciones que originen una nueva población, la cual estará constituida por un conjunto de individuos denominados cromosomas que están compuestos por elementos menores llamados genes, los cuales codifican las instrucciones, parámetros y demás elementos usados para construir una solución al problema por solucionar (Caballero, 2006).

La mayoría de los métodos de AG tienen como componentes: la población de cromosomas (típicamente secuencia de bits), la selección de acuerdo a la aptitud (es un función objetivo) para dar un puntuación a los cromosomas de la población actual, el entrecruzamiento (se producen dos cromosomas recombinados) para producir descendencia y mutaciones aleatorias (cambia alguna posición en el cromosoma) de la descendencia.

La idea general detrás de los AG es imitar la evolución natural (Golberg and Holanda 1988). Una población de soluciones se evalúa con respecto a una función de aptitud. La función de aptitud puede medir el grado en que se cumple el objetivo de aprendizaje (o cualquier otra tarea). La población pasa a través de una serie de generaciones

en el tiempo. El algoritmo comienza con una generación de estrategias de azar construidos. Cada generación se desarrollas a partir de la anterior bajo la influencia de operadores genéticos.

4. RED DE PETRI SIMULADA EN NETLOGO

La RP seleccionada para modelar y simular en NetLogo se presenta en la figura 2. La razón de elección de esta red radica en que es una red compleja, ya que tiene varios tokens de inicio que puede disparar diferentes transiciones y dependiendo de la secuencia de disparo de transiciones, existen diferentes posibles lugares marcados. La red representa una red asimétrica, viva, repetitiva, consistente, no limitada en el lugar 4 (p4) y también no conservativa por ese mismo lugar. Los lugares p2, p4, p5 y p10 tienen un token. La figura mostrada fue exportada desde la herramienta pntool de Matlab, que se utiliza para simular una RP y con la cual es posible comprar los resultados obtenidos.

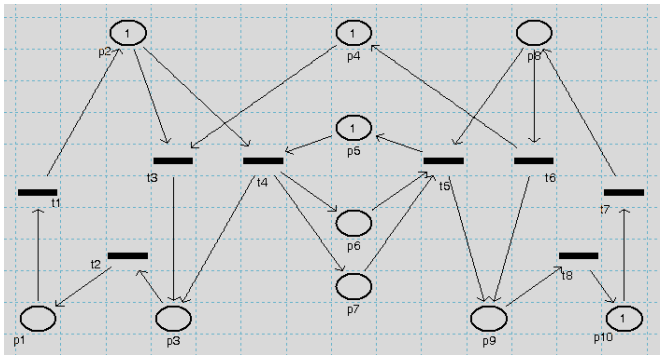


Figure 2: RP seleccionada

4.1 MODELO DE LA RED DE PETRI EN NETLOGO

La RP modelada en NetLogo se representa en la figura 3. Los lugares que no tienen token se representan en color blanco y los que tienen un token se representan en color rojo. Con la idea de usar la representación gráfica de una RP, un lugar se representa con una círculo y una transición se representa con un cuadro.

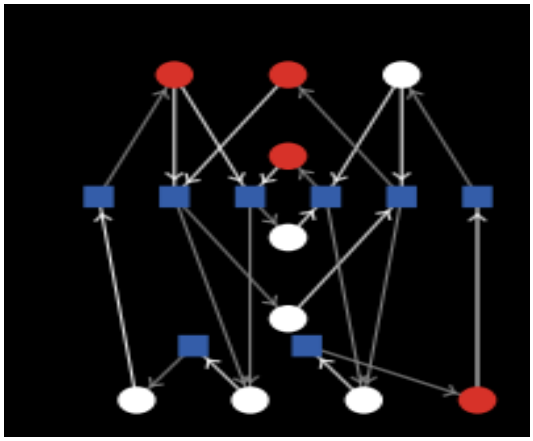


Figure 3: RP modelada en NetLogo

El dibujo mostrado se dejó fijo para que se facilitara comprarla con la red seleccionada, es decir, la colocación de los lugares, transiciones y arcos están establecidos por coordenadas. Los lugares marcados con tokens también se fijan porque se debe tener una posición de inicio de los tokens y se representan con círculos en color rojo. El algoritmo principal implementado en NetLogo para modelar la red se puede ver a continuación.

```
globals [ x0 y0 n l1 l2 l3 t1]

turtles-own [marcada? token]  ;para poder marcar y desmarcar lugares
```

```
breed [lugares1 lugar1]   ; establecidas para las coordinadas y visualización
breed [lugares2 lugar2]
breed [lugares3 lugar3]
breed [transiciones transicion1]
to setup
clear-all
crear-lugares             ;crear los lugares en coordenadas fijas
crear-transiciones        ;crear las transiciones en coordenadas fijas
crear-enlaces-lugares     ;crear los enlaces de los lugares a las transiciones
crear-enlaces-transiciones  ; crear los enlaces de las transiciones a los lugares
inicializar-tokens        ;establecer los marcados iniciales
end
to go
mover-tokens            ; moviendo las marcas (tokens), percibidas en círculos rojos
end
```

4.2 SIMULACIÓN DE LA RED DE PETRI EN NETLOGO

Una vez que se modeló la red la figura 3, se hizo la simulación de la misma. La simulación considera que cuando el token dispara una transición, el lugar marcado en rojo se cambia a color blanco y el que estaba en blanco se cambia a color rojo para indicar que allí está el token.

La simulación de la red se realiza con la marcación (color rojo) y la des-marcación de lugares (color blanco) con base en las transiciones que hay con sus vecinos. Como esta es la parte dinámica y aleatoria de la simulación, se enfatiza y se muestra a continuación el algoritmo:

```
to mover-tokens       ;mover los marcados (tokens) en la red
ask turtles with [marcada?]    ; primero se revisa si el lugar está marcado
[    show who
 desmarcar-lugar
  ask link-neighbors with [token = 0]
    [ask link-neighbors with [color = white]
       [marcar-lugar]]
 desmarcar-lugar ;agregada para que se cambien a blanca   ]
end
to marcar-lugar    ;procedimiento para marcar un lugar  (poner token)
   set token token + 1
   set color red
   set marcada? true
end
to desmarcar-lugar  ; procedimiento para desmarcar un lugar (quitar token)
   set token token - 1
   set color white
   set marcada? false
end
```

5. RESULTADOS

A continuación, en la figura 4, se muestran algunas de las pantallas de la simulación de la RP en NetLogo. El cambio de token de un lugar a otro se ve a través del cambio de color de los lugares de blanco a rojo. Las capturas de imágenes de la simulación se muestran de manera aleatoria. Sin embargo, tanto las capturas de imágenes del NetLogo como las de Matlab guardan el mismo orden de resultados para su comparación.

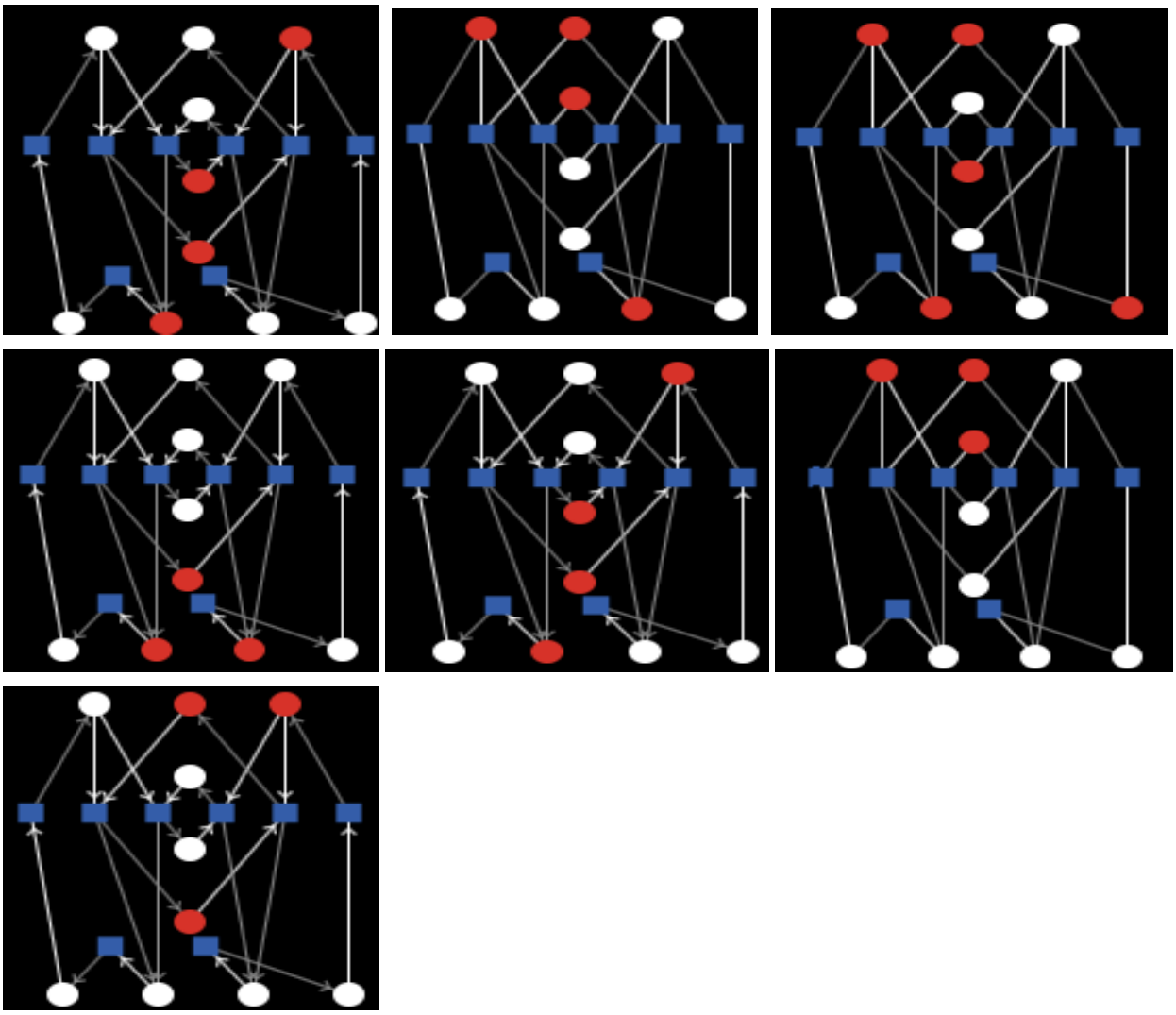
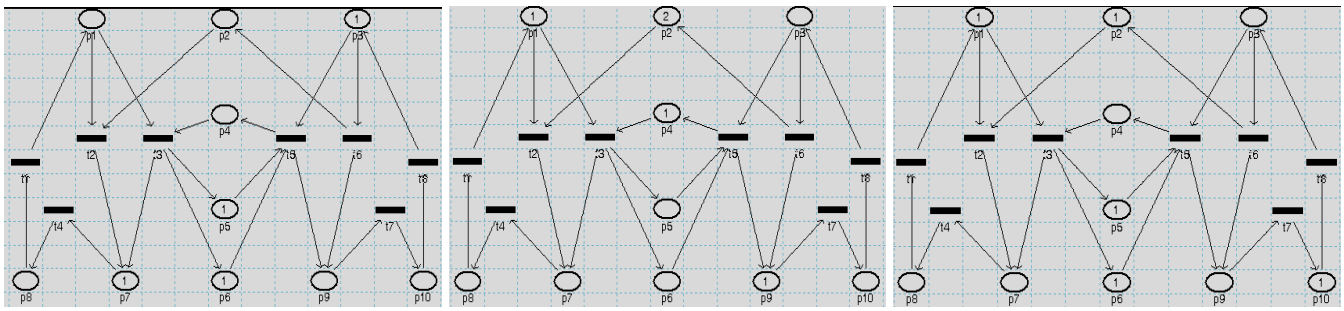


Figure 4: RP simulada en NetLogo

Los resultados con la herramienta pntool de Matlab se pueden ver en la figura 5. Los lugares marcados (los que tienen tokens) se distinguen con un número, que representa el lugar marcado o la indicación de que el lugar contiene token. A diferencia del NetLogo, las transiciones se representan con un rectángulo.



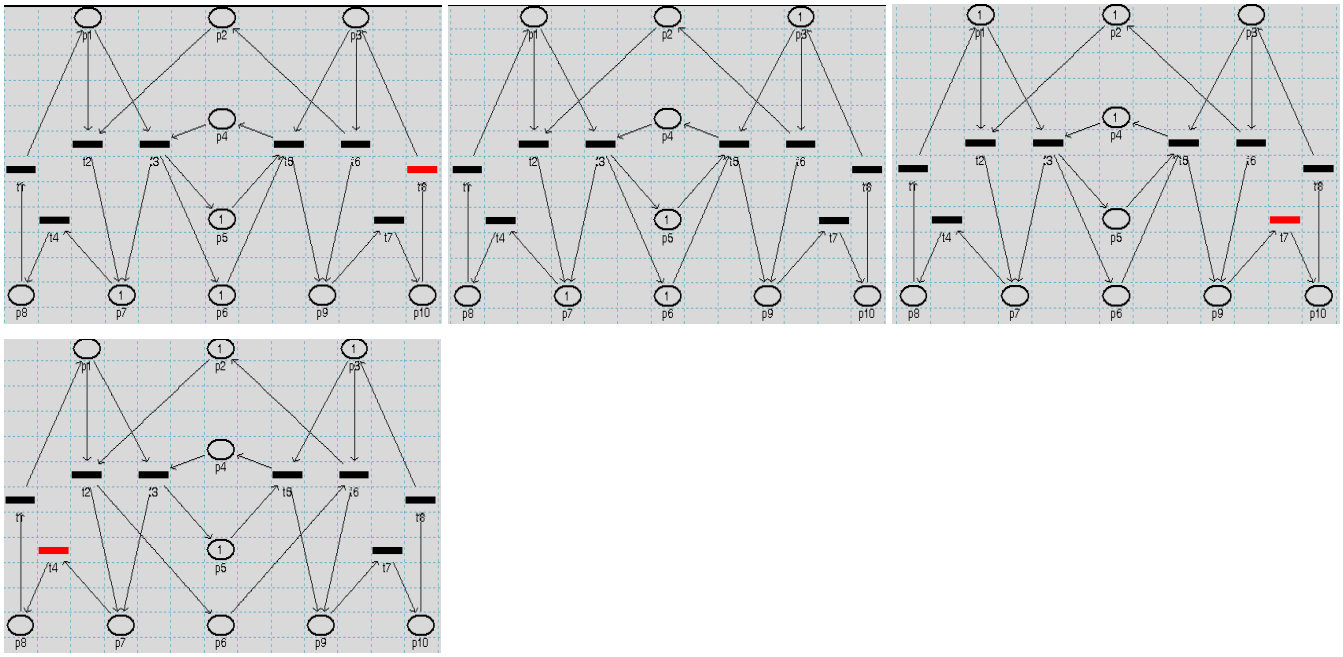


Figure 5: RP simulada en Matlab

Las transiciones marcadas en rojo indican que se va a disparar esa transición. No se muestra en todas las imágenes porque las capturas de imágenes se hicieron en diferentes momentos.

6. CONCLUSIONES Y TRABAJO FUTURO

Si se comparan los resultados de la simulación de la RP en NetLogo con la que se realiza con la herramienta pntool de Matlab se puede ver que son iguales porque los tokens están en los lugares esperados. Sin embargo, una desventaja que se presenta en esta primera etapa de la implementación es que cada vez que se quiera modelar una red se deben establecer las dimensiones fijas en la pantalla de NetLogo. Aunque la ventaja es que una vez que se crea el modelo se puede analizar el comportamiento fácilmente.

El trabajo futuro contempla establecer como propuesta la modificación del algoritmo para que automáticamente genere una RP indicando los lugares y transiciones de la red. Ambos se construyen con agentes denominados tortugas pero con diferente forma así que sería posible generarlas aleatoriamente con las condiciones que se establezcan. Además, las transiciones podrían cambiar de color cuando se activarán para que se notara cuál de ellas se dispara. Con estas modificaciones se podrían modelar sistemas diversos e implementar una función objetivo para minimizar o maximizar los parámetros que se pretendan analizar. Por otra parte, es posible explotar el uso del NetLogo si se quiere optimizar el modelado de la RP a través de una función objetivo si es que se desea optimizar el diseño de la red de cualquier sistema.

REFERENCIAS

Caballero J. P and Gonzalo Mejía. (2006). "Redes de Petri y Algoritmos Genéticos, una propuesta para la Programación de Sistemas de Manufactura Flexible".

Chien C. F. and Chen C. H. (2007). "Using genetic algorithms (GA) and a colored timed Petri Net (CTPN) for modeling the optimization-based schedule generator of genetic production scheduling system". *International Journal of Production Research*, Vol. 45, No. 8. Pp. 1763-1789.

Lihong. Q., Yixin Z. Jianjun Y. and Yang L. (2009). "A Petri Net and Genetic Algorithm Based Method for Flexible Manufacturing Cells Modeling and Scheduling". *Trans Tech Publications.* Doi:10.4028/www.scientific.net/KEM.407-408.268. Vols 407-408 pp. 268-272.

Marwan W., A. Wagler and R. Weismantel. (2007). "A mathematical approach to solve the network reconstruction problem". *Sringer-Verlag*. Vol. 67, pp. 117-132.

Mayo M. and Beretta L. (2010). "Modelling Epitasis in Genetic Disease using Petri Nets, Evolucionary Computation and Frequent Itemset Mining". *Expert Systems with Applications*. Doi: 10.1016/j.eswa.2010.09.062

Mauch H. (2003). "Evolving Petri Nets with a Genetic Algorithm". *Springer-Verlag Berlin Heidelberg.* pp.1810-1811.

O'Neill M. R, Allan V.H. Flann N. and Chen H. (2012). "Petri Net Representation for Parallel Loop Scheduling Using a Genetic Algorithm". pp. 1-11 . *http://citeseerx.ist.psu.edu/* . 03/02/2012.

Prashant J. Kummanan S. and Krischanaiah O.V. (2001). "Application of Petri Nets and a Genetic Algorithm to Multi-Mode-Resource Constrained Poject Scheduling". *The International Journal of Advanced Manufacturing Technology.* Vol. , No. 17, pp 305-314.

Reid D. J. (1998). "Constructing Petri Net Models Using Genetic Search". Mathematical and Computer Modelling, Elsevier. Vol. 27, No. 8. Pp. 85-103.

Sandqvist S. (2002)."Aspects of Modelling and Simulation of Genetic Algorithms: A Formal Approach". Research Reports 74. Edition, Helsinki Univertisy of Technology. Esbo, Finland.

Silva M. (1985). *Las redes de Petri: en la automática y la informática*. 1ra edición, AC, Madrid, España.

Wolters B. and Steffens T. (2008). "Learning Agent-Behavior for Agent-Based Simulation Using Genetic Algorithms". pp. 1-5

Duan, L., Loh, J.T., and Chen, W.F. (1990). "M-P-F based analysis of dented tubular members". *Journal of Structural Engineering*, Vol. 21, No. 8, pp 34-44.

Fang, T.C. (1987). "Network resource allocation using an expert system with fuzzy logic reasoning", Ph.D. thesis, University of California at Berkeley, California, USA.

Hong Kong MTR Corporation. (2001). Passenger Data for 1990-2000, http://www.mtr.com.hk, mm/dd/yy. (date accessed)

Paulson, B.C., and Barrie, D.S. (1992). *Professional Construction Management*, 3rd edition, Mcgraw-Hill International, Singapore.

Peter, J. (1998). "Development of a risk management model for international joint ventures", *Proceedings of Second International Conference on Project Management,* Editors: L.R.K. Tiong, National University of Singapore, Singapore, pp. 55-67.

Truman, H. (1990). Private Communications.

Van Hoover, M. (2002). Interview, 7 August 2002.

Autorización y Renuncia

Los autores autorizan a LACCEI para publicar el escrito en las memorias de la conferencia. LACCEI o los editores no son responsables ni por el contenido ni por las implicaciones de lo que esta expresado en el escrito