RESOLUCIÓN DE CONFLICTOS EN LA PROGRAMACIÓN DE ACTIVIDADES DE SISTEMAS PRODUCTIVOS UTILIZANDO COLORACIÓN DE GRAFOS

SCHEDULING ACTIVITIES WITH CONFLICT RESOLUTION IN PRODUCTIVE SYSTEMS USING GRAPH COLOURING

RESUMEN
En este artículo se expone un método formal para la programación de actividades que presentan conflictos por utilización simultánea de recursos. Se muestra como mediante la construcción del modelo de un sistema productivo usando grafos y la posterior aplicación de un método de coloreo de nodos, es posible elaborar una programación de tareas que pueden ser llevadas a cabo sin impedimentos. El método presentado se ilustra mediante un ejemplo representativo de las tareas de mantenimiento de un sistema mecánico, igualmente se presenta un algoritmo que permite optimizar el número de iteraciones necesarias para el coloreo de nodos de un grafo.

PALABRAS CLAVES: Grafos Coloridos, Mantenimiento, Resolución de Conflictos, Sistemas Productivos.

ABSTRACT
This paper describes a formal method for programming activities that present conflicts derived from simultaneous use of resources. We show the model making of a production system using graphs and the subsequent use of a method of coloring nodes, the result shows how is possible to achieve scheduling tasks that can be carried out without impediment. The method presented is illustrated using a representative example of the maintenance activities of a mechanical system. An algorithm that allows optimizing the number of iterations required for coloring nodes in a graph is also presented.

KEYWORDS: Coloured Graphs, Conflict solution, Maintenance, Productive Systems.

PEDRO L. ANGEL RESTREPO
Ingeniero Mecánico, M. Sc.
Profesor Asistente
Universidad de Antioquia
plangel@udea.edu.co

PAOLA ANDREA DEL RIO RUIZ
Estudiante Programa de Bioingeniería
Universidad de Antioquia
paolaandreadelrio@udea.edu.co

CAROLINA VELEZ ORTEGA
Estudiante Programa de Bioingeniería
Universidad de Antioquia
carolinavelez@udea.edu.co

Grupo de **D**esarrollo e **I**nvestigación en **S**istemas **I**nformáticos y **D**iseño **E**lectrónico aplicado a las **N**uevas **T**ecnologías en **S**alud, ***DISIDENTES***, Facultad de Ingeniería, Universidad de Antioquia.

1. INTRODUCCIÓN

La complejidad de los sistemas productivos construidos por el hombre se incrementa día a día, es por ello que se hace necesario contar con procedimientos que permitan la planificación formal de las actividades necesarias para poder producir. Una de las situaciones mas frecuentes en la que se hace necesario elaborar una programación de actividades es en el mantenimiento de la infraestructura de producción, este proceso se caracteriza por la necesidad de llevar a cabo un conjunto de tareas en las cuales se utilizan recursos técnicos y humanos, procurando obtener una programación eficaz y eficiente.

El proceso de planificación del mantenimiento es una tarea altamente compleja [1], en la cual se presenta de forma habitual que algunas actividades no puedan ser programadas de manera simultánea ya que requieren de la utilización de un mismo factor, y en este sentido se dice que presentan un conflicto por utilización simultánea de recursos. Por otro lado, la ejecución puntual de la programación de las tareas de mantenimiento suele ser difícil de llevar a cabo, ya sea por que se presenten retrasos en la realización de las actividades o por la ocurrencia de imprevistos que exigen intervenciones de tipo correctivo y que originan cambios en la planeación previa, obligando de esta manera a una reformulación de la misma. Por tal razón cobra importancia la necesidad de contar con métodos algorítmicos para planificar de forma automática las actividades de mantenimiento [2].

Dicha necesidad de lograr la programación de actividades de forma rápida y eficiente, al igual que la ventaja ofrecida por los procedimientos que pueden ser implementados en herramientas computacionales, hacen de los métodos formales una herramienta conveniente y adecuada para la gestión de los procesos de administración de la infraestructura de los sistemas productivos. Una técnica formal con creciente aceptabilidad en el área de planificación de actividades de procesos productivos es el modelado con grafos [3], específicamente el coloreo de los mismos [4], [5].

Fecha de Recepción: 26 de enero de 2009
Fecha de Aceptación: 28 de marzo de 2009

En este orden de ideas, este trabajo presenta un método computacional por el cual, mediante la representación de las actividades de un sistema productivo utilizando grafos y la aplicación del coloreo de vértices, es posible obtener subconjuntos de tareas exentos de conflictos permitiendo obtener programaciones adecuadas de las actividades que se deben llevar a cabo, presentando de esta manera una herramienta útil en la administración de los procesos productivos. El artículo está organizado como sigue: en la segunda parte se presentan los principios básicos de la teoría de grafos, en la tercera se muestra como las tareas de mantenimiento de un sistema productivo pueden ser representadas por un grafo, en la cuarta parte se introduce un algoritmo para el colorido de grafos y en la quinta se aplica el mencionado algoritmo para resolver el problema de la planificación de las tareas de mantenimiento. Finalmente se exponen las conclusiones y recomendaciones.

2. INTRODUCCION A LA TEORÍA DE GRAFOS

Un grafo se define como una estructura matemática compuesta por dos conjuntos cuyos elementos están relacionados entre sí por una función de asociación. Desde un punto de vista formal un grafo es un conjunto: $G=\{V,A\}$ tal que los elementos de V se denominan nodos o vértices mientras que los elementos de A se denominan arcos o aristas. La representación geométrica de un grafo se construye a partir de círculos para los nodos y líneas para los arcos, cuando la relación establecida por los arcos se da de un nodo a otro de manera no recíproca, el arco recibe el nombre de arco dirigido y se representa por una línea terminada en una flecha que apunta hacia el nodo donde la relación se cumple. Los arcos tienen una función que asocia un valor al arco, dicho valor se conoce con el nombre de peso, si los arcos de un grafo dado tienen el mismo peso usualmente se omiten en su representación geométrica, al número de arcos incidentes a un vértice se le denomina orden del nodo. La figura 1 ilustra pictóricamente un grafo.

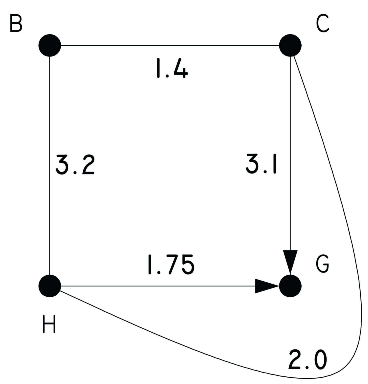


Figura 1. Representación pictórica de un grafo.

La representación matricial de un grafo se realiza, básicamente, por medio de dos estructuras: La matriz de Incidencia (I) y la de Adyacencia (Ad). La matriz de Incidencia relaciona los arcos con los nodos, de tal manera que en las entradas donde se interceptan la fila asociada a los vértices con la columna que representa el arco, se registra el valor del peso del arco, en el caso de grafos no pesados el valor de la entrada es uno (***1***). Por otro lado, la matriz de adyacencia relaciona los nodos entre sí, de tal forma que si existe un arco entre dos nodos la entrada correspondiente a la columna y fila de los mismos tiene un valor de uno (***1***) [6]. Las matrices de incidencia y adyacencia del grafo presentado en la figura 1 se ilustran en la figura 2a y 2b respectivamente.

$$I = \begin{array}{c} \\ B \\ C \\ G \\ H \end{array} \begin{array}{c} \begin{array}{ccccc} BC & BH & CG & CH & HG \end{array} \\ \begin{bmatrix} 1.40 & 3.20 & 0.00 & 0.00 & 0.00 \\ 1.40 & 0.00 & 0.00 & 2.00 & 0.00 \\ 0.00 & 0.00 & 3.10 & 0.00 & 1.75 \\ 0.00 & 3.20 & 0.00 & 2.00 & 0.00 \end{bmatrix} \end{array}$$

2.b Matriz de Incidencia

$$Ad = \begin{array}{c} \\ B \\ C \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccc} B & C & G & H \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

2.a Matriz de Adyacencia

Figura 2. Ejemplo de representación matricial de un grafo

3. REPRESENTACIÓN POR GRAFOS DE LAS ACTIVIDADES DE UN SISTEMA PRODUCTIVO

Es posible representar las actividades de los sistemas productivos mediante grafos [7], [8]. En el caso específico de las tareas de mantenimiento, el grafo se construye representando a las actividades por nodos y cuando dos actividades requieran de al menos un recurso en común para ser llevadas a cabo se trazan un arco entre ellas. De esta manera la incompatibilidad por el uso simultáneo de recursos queda registrada en la estructura de datos utilizada para modelar las actividades del sistema de mantenimiento, esto es, el grafo. Como la relación de incompatibilidad es mutua el arco es no dirigido, por simplicidad se asume que la duración de las tareas de mantenimiento es la misma, para la solución de problemas donde sea necesario contemplar diferentes tiempos de duración se requiere considerar multicolorido de vértices [9], [10].

En la tabla 1, se ejemplifican actividades de mantenimiento que se presentan de manera típica en la programación de una jornada de trabajo, detallando los recursos necesarios para que sean llevadas a cabo. El grafo que ilustra la relación de actividades y recursos, expresando las relaciones de incompatibilidad se presenta en la figura 2. La representación matricial del grafo de incompatibilidades se presenta en la figura 3.

Recurso	A_1	A	A	A	A	A	A	A	A	A_1	A_1	A_1	A_1	A_1	A_1
Mecánico 1	▱			▱		▱					▱			▱	
Mecánico 2	▱		▱		▱		▱	▱					▱		▱
Electricista		▱				▱		▱		▱		▱		▱	
Técnico electrónico		▱			▱		▱		▱				▱		
Caja de Herramientas 1	▱			▱		▱		▱						▱	
Caja de Herramientas 2		▱									▱	▱	▱		
Equipo de Limpieza 1	▱		▱							▱		▱		▱	
Equipo de Limpieza 2				▱			▱								▱
Sistema de Calibración		▱			▱		▱		▱						

Tabla 1. Relación recursos necesarios y actividades de mantenimiento.

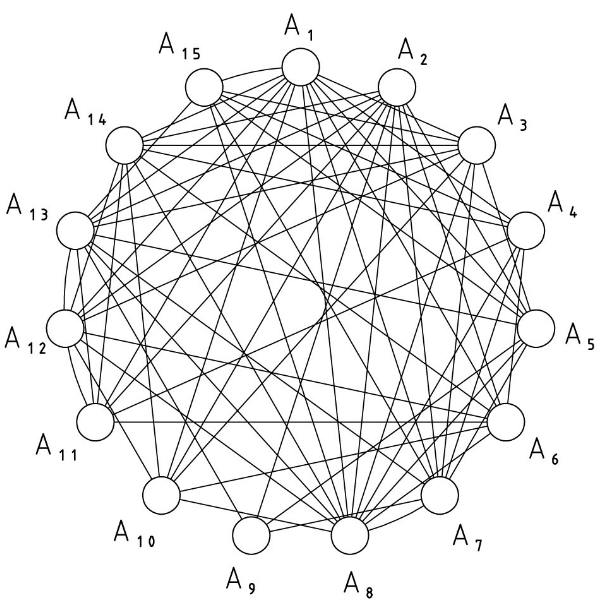


Figura 2. Grafo de Incompatibilidades de las tareas de mantenimiento

$$
\begin{array}{c|ccccccccccccccc}
 & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} & A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\
\hline
A_1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
A_3 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
A_4 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
A_5 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
A_6 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
A_7 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
A_8 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
A_9 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
A_{10} & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
A_{11} & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
A_{12} & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
A_{13} & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
A_{14} & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
A_{15} & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{array}
$$

Figura 3. Representación por medio de la matriz de adyacencia del grafo de incompatibilidades de las tareas de mantenimiento

De esta manera, el problema de obtener los conjuntos de tareas que puedan ser realizadas de manera simultánea se reduce a obtener subconjuntos de nodos que no sean adyacentes, es decir que no contengan vértices entre los cuales exista un arco. Este problema se conoce en la teoría de grafos como el coloreo de nodos [11], en el apartado siguiente se presenta un algoritmo para el coloreo de los nodos de un grafo.

4. ALGORITMO PARA EL COLOREO DE NODOS DE GRAFOS

Existen múltiples algoritmos para el coloreo de nodos de grafos entre los cuales se destacan los mencionados en: [12], [13] y [14]. El método computacional presentado en la figura 4, utiliza la matriz de adyacencia del grafo de incompatibilidades como estructura de datos de entrada. El algoritmo tiene la ventaja de partir del nodo de mayor orden al de menor, permitiendo disminuir el número de operaciones necesarias para asignar los colores.

Uno de los resultados esperados del coloreo de grafos es el número cromático, el cual denota la cantidad de subconjuntos de nodos en los que el conjunto original de vértices (V) puede ser dividido de tal manera que los elementos de un subconjunto determinado no posean arcos entre sí. A cada subconjunto de vértices se le asocia un color (representado usualmente por un número entero), como forma de diferenciar sus elementos de los pertenecientes a los de los otros subconjuntos. En el contexto de los sistemas productivos cada color significa un conjunto de actividades independientes de conflictos por utilización de recursos, y por tanto aquellos grupos de actividades que pueden ser llevadas a cado sin impedimentos.

Entrada: Matriz $M = [A, A]$

donde: $|A| = N$, cardinalidad del conjunto de vertices (V) del grafo de Incompatibilidades

Retorno: Número cromático del grafo: C_{max}
Lista de subconjuntos de nodos: $L = \{l_1, l_2, \cdots l_n\}$

donde: $\{l_1 \cup l_2 \cdots \cup l_n = V\}$ **y** $\{l_1 \cap l_2 \cdots \cap l_n\} = \phi$

Inicio

$//$ *Encontrar el grado de cada nodo*

Para $i = 1$ ***hasta*** N **hacer**
  Para $j = 1$ ***hasta*** N **hacer**
    $ArregloNodos\,[i]\,.GradoNodo = M[i, j] +$
    $ArregloNodos\,[i]\,.GradoNodo$
  $ArregloNodos\,[i]\,.PosicionNodoMatriz = i$

$//$ *Ordenar los nodos de mayor a menor grado*

$ArregloNodos\,[] =$ **Quicksort**$(ArregloNodos\,[]\,.GradoNodo)$

$//$ *Asignar colores a los vértices del grafo*

Para $i = 1$ ***hasta*** N **hacer**
  Si $ArregloNodos\,[i]\,.ColorNodo] = \phi$ **Entonces:**
    $ArregloNodos\,[i]\,.ColorNodo] = C_1$
  $ArregloNodos\,[i]\,.TipoColorNodo = Definitivo$
  Para $j = 1$ ***hasta*** N **hacer**
    Si $M[ArregloNodos\,[i]\,.PosicionNodoMatriz, j] > 0$ ***Y***
    $ArregloNodos\,[j]\,.TipoColorNodo$
    $<>$ $Definitivo$ **Entonces:**
      Si $ArregloNodos\,[i]\,.ColorNodo <> C_1$ **Entonces:**
        $ArregloNodos\,[j]\,.ColorNodo] = C_{i+1}$
        Si $C_{max} <$
        $ArregloNodos\,[j]\,.ColorNodo$ **Entonces:**
          $C_{max} = ArregloNodos\,[j]\,.ColorNodo$
      Si no:
        $ArregloNodos\,[j]\,.ColorNodo] = C_1$
      $ArregloNodos\,[j]\,.TipoColorNodo = Temporal$

$//$ *Generar listas de nodos agrupados por color*

Para $i = 1$ ***hasta*** N **hacer**
  $ListaNodosColor[ArregloNodos\,[i]\,.ColorNodo] =$
  Adicionar$(ArregloNodos\,[i])$

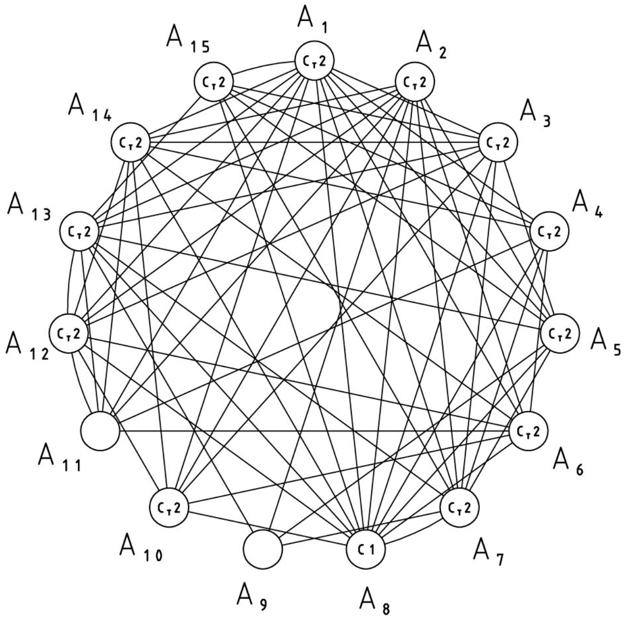
Fin

Figura 4. Algoritmo de coloración de nodos de un grafo.

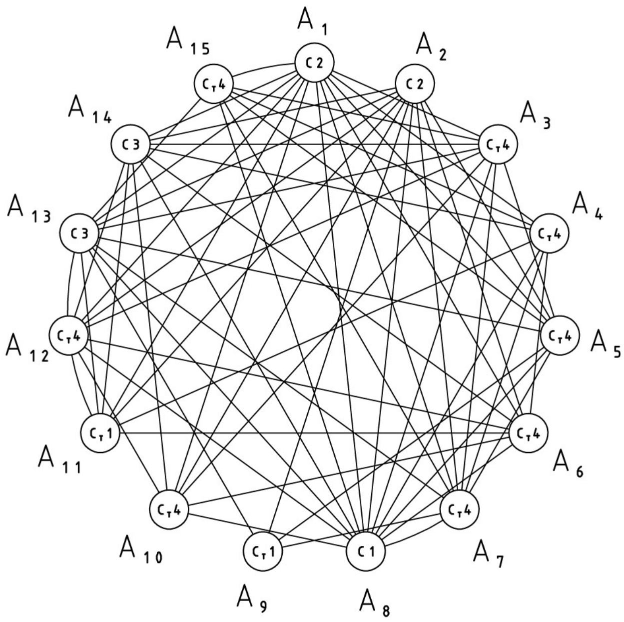
5. PLANIFICACIÓN DE TAREAS DE MANTENIMIENTO

Aplicando el algoritmo presentado en la figura 4, teniendo como datos de entrada la matriz presentada en la figura 3, se obtiene como número cromático del grafo el número siete (*7*). Las iteraciones uno (1), cinco (5), diez (10) y quince (15), del algoritmo se presentan en las figuras 5.a, 5.b, 5.c y 5.d respectivamente. La agrupación de tareas que arroja el algoritmo se muestra en la tabla 2.

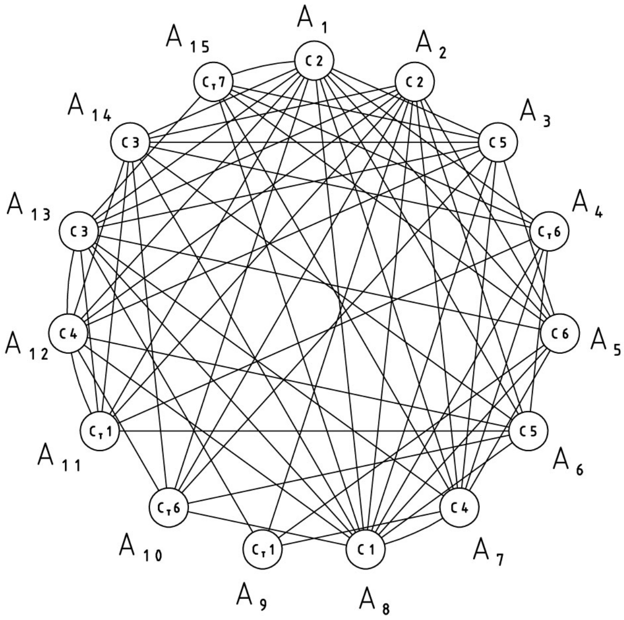
Los resultados arrojados por el algoritmo consisten en agrupaciones de nodos, representando las actividades que pueden ser llevadas a cabo simultáneamente pues no compiten por la utilización de recursos. La solución arrojada por el algoritmo no es única en tanto que a los conjuntos de tareas se les puede asignar diferente orden de ejecución, de hecho cada posible combinación de los subconjuntos es una secuencia valida de actividades. Además en determinados casos los elementos de algunos subconjuntos pueden ser intercambiados, sin embargo la solución es óptima en el sentido que el número cromático, que en este caso representa el número mínimo de unidades de tiempo en que las actividades pueden ser realizadas es una propiedad estructural del grafo y es único.



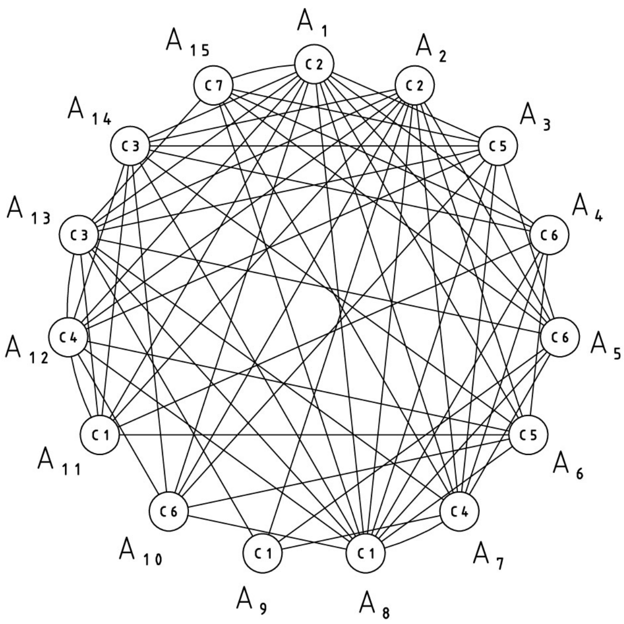
5.a Primera iteración



5.b Quinta iteración



5.c Décima iteración



5.d Décimo quinta iteración

Figura 5. Representación grafica de diversas iteraciones del algoritmo de coloreo de nodos

Con todo, la solución presentada en la tabla 2. se obtiene aplicando un método heurístico para disminuir las iteraciones del proceso de colorido del grafo, y por lo tanto el tiempo de ejecución computacional. Dicha heurística consiste en iniciar asignando color de los nodos de mayor a los de menor orden, evitando la verificación redundante de conexiones entre los nodos.

Color	Elementos de los subconjuntos de nodos	Número de Recursos Utilizados por el grupo
C_1	A_8, A_9, A_{11}	7
C_2	A_1, A_2	8
C_3	A_{13}, A_{14}	7
C_4	A_7, A_{12}	7
C_5	A_3, A_6	5
C_6	A_4, A_5, A_{10}	8
C_7	A_{15}	2

Tabla 2. Vértices agrupados por colores.

Estos subgrupos de actividades pueden ser programados en cualquier arreglo, no necesariamente en el orden del valor asociado al color. Para este caso específico se tienen $(n-1)! = 6! = 720$ posibles combinaciones. Una selección de la programación óptima, consiste en evitar que un retraso en la ejecución de un subconjunto de actividades se propague a los siguientes, para lo cual se aplica una heurística que minimiza la adyacencia de los subconjuntos de nodos a ser recorridos en el grafo.

Progra-mación	Subconjuntos de Actividades Agrupadas por Colores
1	$C_2 \to C_7 \to C_6 \to C_5 \to C_1 \to C_3 \to C_4$
2	$C_2 \to C_7 \to C_6 \to C_5 \to C_1 \to C_4 \to C_3$
3	$C_2 \to C_7 \to C_6 \to C_5 \to C_3 \to C_4 \to C_7$
4	$C_2 \to C_7 \to C_6 \to C_5 \to C_3 \to C_7 \to C_4$
5	$C_2 \to C_7 \to C_6 \to C_5 \to C_4 \to C_1 \to C_3$
6	$C_2 \to C_7 \to C_6 \to C_5 \to C_4 \to C_3 \to C_2$
7	$C_6 \to C_7 \to C_2 \to C_5 \to C_1 \to C_3 \to C_4$
8	$C_6 \to C_7 \to C_2 \to C_5 \to C_1 \to C_4 \to C_3$
9	$C_6 \to C_7 \to C_2 \to C_5 \to C_3 \to C_4 \to C_1$
10	$C_6 \to C_7 \to C_2 \to C_5 \to C_3 \to C_1 \to C_4$
11	$C_6 \to C_7 \to C_2 \to C_5 \to C_4 \to C_1 \to C_3$
12	$C_6 \to C_7 \to C_2 \to C_5 \to C_4 \to C_3 \to C_1$

Tabla 3. Programaciones de actividades.

Esto se logra al intercalar los conjuntos de actividades de mayor utilización de recursos con los de menor, en tanto que en algunos subgrupos se utilizan el mismo número de recursos, se tienen varias opciones de programación como se muestra en la tabla 3. La selección de una de estas programaciones requiere de conocimiento especifico del sistema, por ejemplo de la confiabilidad de las maquinas y/o disponibilidad de suministros. En tanto que en el grafo de incompatibilidades el uso simultáneo de recursos se representa por arcos, la posibilidad de utilizar colorido de arcos en conjunto con el colorido de nodos es sumamente atrayente como fuente de soluciones al problema de encontrar programaciones óptimas de actividades en sistemas productivos.

6. CONCLUSIONES Y RECOMENDACIONES

La estructura de datos que representa el grafo de incompatibilidades es una descripción sencilla, compacta y funcional de los conflictos por uso de recursos que se presentan en la programación de labores en sistemas productivos. Dicha estructura permite obtener información sobre aspectos críticos en la solución de conflictos por utilización simultánea de recursos.

El método de coloreo de grafos prueba ser una herramienta válida y adecuada para la programación de actividades de sistemas productivos, donde se requiera la solución de conflictos por utilización simultánea de recursos. Sin embargo criterios específicos relacionados con el desempeño del sistema productivo que se quiere administrar son necesarios para obtener programaciones de mantenimiento óptimas.

La conveniencia de utilizar métodos formales en problemas donde las tareas requieran tiempos de ejecución diferentes, apunta a que la utilización de métodos avanzados de la teoría de grafos, tales como el multicoloreo de nodos y el coloreo de arcos, puedan ser utilizados como técnicas de gran impacto en el área de planificación de actividades de sistemas productivos.

Igualmente, el estudio de técnicas de recorridos en los grafos de incompatibilidades ofrece fuentes de estudio atractivas para encontrar respuestas óptimas al problema de ocurrencia de retrasos en serie, entre subconjuntos de tareas de mantenimiento, y en general en la planificación de actividades en sistemas productivos.

7. BIBLIOGRAFÍA

[1] Y. Meng-Lai and. Arellano R. R., "The Importance of Maintenance Planning – A Case Study," in *Proc*. 2007 *Reliability and Maintainability Symposium, RAMS '07*. pp. 482-487

[2] S. L. Dreyer, "Advance Maintenance Planning and Schedule," in *Proc.* 2006 Autotestcon, pp. 341-347

[3] S. Irany, V. Leung, "Scheduling with Conflicts on Bipartite and Interval Graphs," *Journal of Scheduling*, *Special issue: On-line Scheduling*. vol 6, no 3, pp. 287-307, 2003.

[4] N. Zufferey, P. Amstutz, Ph. Giaccari, "Graph colouring approaches for a satellite range scheduling problem," *Journal of Scheduling*. vol 11, pp. 263-277, 2008.

[5] K. Jansen, L. Porkolab, "Preemptive Scheduling with Dedicated Processors: Applications of Fractional Graph Coloring," *Journal of Scheduling*. vol 7, pp. 35-48, 2004.

[6] A. Gibbons, *Algorithmic Graph Theory*, New York: Cambridge University Press, 1985, p. 17.

[7] F. Junqueira, P. E. Miyagi, "A New Method for the Hierarchical Modeling of Productive Systems," in *Information Technology for Balanced Manufactiuing Systems, IFIP*, vol. 220, Ed. Shen, 2006, pp. 479-488.

[8] M. Silva, E. Teruel, R. Valette and H. Pingaud, "Petri Nets and Production Systems," in *Lectures on Petri Nets II: Applications, LNCS* vol. 1492/1998, Springer Berlin / Heidelberg, 1998, pp. 85-124.

[9] M. M. Halldórsson and G. Kortsarz, "Multicoloring: Problems and Techniques," *in Mathematical Foundations of Computer Science*, *LNCS* vol. 3153/2004, Springer Berlin / Heidelberg, 2004, pp. 25-41.

[10] D. Marx, "Graph coloring problems and their applications in scheduling," in *Proc*. *John von Neumann PhD Students Conference*. Budapest, 2003.

[11] G. Chartrand, P. Zhang, *Introduction to Graph Theory*, New York: McGraw Hill, 2005, pp.267-268.

[12] M. D. Smith, N. Ramsey and G. Holloway, "A Generalized Algorithm for Graph-Coloring Register Allocation," in *Proc 2004 ACM SIGPLAN Programming language design and implementation Conf. on*, pp. 277-288.

[13] A. Koseki, H. Komatsu and T. Nakatani, "Preference-directed graph coloring," SIGPLAN Notices., vol. 37, No. 5 pp. 33-44, 2002.

[14] A. Lim, Y. Zhu, Q. Lou and B. Rodrigues, "Heuristic Methods for Graph Coloring Problems," in: *Proc. 2005 ACM symposium on Applied computing Conf. on,* pp. 933-939.