

# REPRESENTACIÓN GRÁFICA DEL PROBLEMA DE MÁQUINAS EN PARALELO NO RELACIONADAS PARA COLONIA DE HORMIGAS POR MEDIO DE UN GRAFO DISYUNTIVO.

Alina Martínez Oropeza

CIICAp, Universidad Autónoma del Estado de Morelos  
Av. Universidad 1001, Chamilpa, 62209, Cuernavaca Morelos, MÉXICO  
alinam@uaem.mx

**Resumen.** El presente trabajo propone un modelado del problema de Máquinas en Paralelo no Relacionadas para Colonia de Hormigas por medio de un grafo disyuntivo, el cual facilita la comprensión del comportamiento del método de solución aplicado al problema antes mencionado. Para entender el funcionamiento del grafo propuesto, se realizó un análisis por medio de un digrafo para una solución particular a una instancia pequeña.

Se da una introducción a definiciones básicas de teoría de grafos. Se analizan las características básicas del problema de Máquinas en Paralelo no Relacionadas, y se da una introducción general a Colonia de Hormigas, lo que da las bases para analizar las características tanto del problema como del método de solución, para lograr un modelo de grafos eficiente que permita mejorar la comprensión de Colonia de Hormigas aplicado a un problema de Calendarización de tipo NP.

## I. Introducción

En la actualidad existen dentro de la industria gran cantidad de problemas que requieren solución, debido a que conllevan a las empresas al uso poco eficiente de los recursos, lo que a su vez involucra un gasto excesivo y grandes pérdidas de tiempo. Estos problemas han sido abordados con diversos métodos por un sinnúmero de investigadores alrededor del mundo buscando obtener mejoras a las cotas conocidas en la literatura. Uno de los problemas más estudiados en esta área, es el Problema de Calendarización de Trabajos en Talleres de Manufactura; JSSP por sus siglas en inglés (*Job Shop Scheduling Problem*); el cual es considerado parte medular de la industria, debido a que forma parte de las líneas de producción de diversas empresas a nivel mundial. De éste se deriva el problema de Máquinas en Paralelo no Relacionadas; UPMP por sus siglas en inglés (*Unrelated Parallel Machines Problem*). Este problema ha sido estudiado por gran cantidad de investigadores debido a que mantiene características de su antecesor, relajando algunas otras, lo que permite abordar el problema de una manera más sencilla.

Muchos de los problemas con los que se tiene contacto en la vida diaria, pueden ser expresados por medio de modelos matemáticos que involucren el uso de una función objetivo, o bien representados por medio de grafos para facilitar su entendimiento y M. A. Cruz-Chávez, M. H. Cruz-Rosales, J. del C. Peralta- Abarca (Eds): Jornadas de Vinculación Académica. pp. 24 – 39. 2011.

modelado por métodos computacionales, para ser tratados por la Optimización Combinatoria. La *Optimización Combinatoria* es una rama de gran importancia dentro de las ciencias computacionales, dedicada a la investigación de operaciones, así como al estudio y tratamiento de problemas considerados difíciles de resolver [1].

En este trabajo se realiza un análisis tanto de las restricciones del problema UPMP como de Colonia de Hormigas. Este análisis se toma como base para modelar el problema por medio de un grafo disyuntivo, tomando en cuenta las restricciones básicas del problema así como del método de solución aplicado, explicando ampliamente un digrafo propuesto para una solución particular.

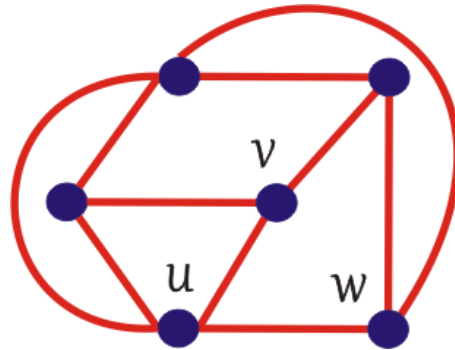
El presente artículo se divide en las siguientes secciones: la sección II presenta una explicación de los conceptos básicos de grafos y digrafos, la sección III explica las características del problema de Máquinas en Paralelo no Relacionadas, la sección IV da un panorama general del Colonia de Hormigas, la sección V explica ampliamente una representación del UPMP para Colonia de Hormigas por medio de un grafo disyuntivo, para representar una solución particular al UPMP (clasificado como un problema de tipo NP) por medio de un digrafo, para finalmente mostrar las conclusiones de este trabajo de investigación en la sección VI.

## II. Grafos y Digrafos

Un *grafo* es una estructura combinatoria de gran importancia, y por tanto, ampliamente utilizada en las ciencias computacionales [14], que permite expresar de forma sencilla y efectiva a través de modelos, las relaciones existentes entre elementos de uno o más conjuntos. Estos modelos se pueden definir de manera general como  $G = (V, E)$ , el cual consta de dos subconjuntos finitos y no vacíos, un subconjunto  $V = V(G)$  de elementos denominados *vértices* y un subconjunto  $E = E(G)$  de pares no ordenados de vértices distintos llamados *aristas*. Si la arista  $e = \{u, v\} = uv$  empareja los vértices  $u$  y  $v$ , se dice que  $u$  y  $v$  son vértices adyacentes, además de que la arista  $e$  es incidente tanto en  $u$  como en  $v$  [2]. De lo contrario, se dice que se trata de vértices *independientes*. Para evitar ambigüedades, se debe asumir que  $V \cap E = \emptyset$  [3].

Un grafo cuenta con un orden y un tamaño, donde el *orden* está dado por el número de vértices de  $G$ ,  $|V(G)|$  y su *tamaño* se encuentra representado por el número de total de aristas  $|E(G)|$  existentes en el grafo.

Los subconjuntos de un grafo son representados de manera general con puntos para los *vértices*, los cuales son unidos por líneas denominadas *aristas* que convierten a los vértices emparejados en *vértices adyacentes*. En la figura 1 se muestra un ejemplo de la representación general de un grafo de orden 6 y tamaño 10.



**Fig. 1.** Grafo  $G = (V, E)$  de orden 6 y tamaño 10.

De modo que de acuerdo a la figura 1, es posible observar que los vértices  $u$  y  $v$  son adyacentes, mientras que  $u$  y  $w$  se muestran como vértices independientes.

Los grafos pueden ser clasificados de manera general como *grafos dirigidos* y *grafos no dirigidos*.

Formalmente, un *grafo no dirigido* cuenta con las siguientes características:

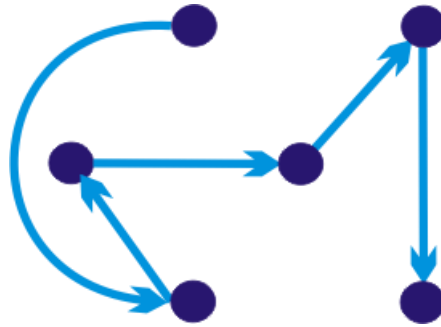
1. Un conjunto finito de vértices  $V$ .
2. Un conjunto de aristas  $E$  en el que cada arista es un emparejamiento de dos vértices.

Un *grafo dirigido* (*digrafo*) cuenta con:

1. Un conjunto finito de vértices  $V$ .
2. Un conjunto de arcos  $E \subset V \times V$ , tomando en cuenta que cada arco es un par ordenado de vértices.

## Digrafos

Un *digrafo* es un grafo dirigido donde las *aristas* (arcos) son pares ordenados de vértices distintos; de modo que un digrafo  $G = (V, A)$  es una estructura combinatoria formada por un par  $(V, A)$  de conjuntos disjuntos. Es decir, al representar una solución particular a un grafo, los arcos disyuntivos representados toman una orientación, cambiando a ser arcos conjuntivos (Figura 2).



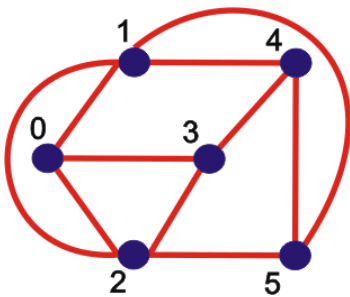
**Fig. 2.** Digrafo representando una posible solución

Al ser un grafo o digrafo una representación gráfica de un problema combinatorio, se requiere de una representación matricial para ser procesado computacionalmente. Dos de las representaciones más comúnmente utilizadas son la *Matriz de Adyacencia* y la *Matriz de Incidencia* [2].

- *Matriz de Adyacencia:* Es una matriz simétrica  $n \times n$ , que muestra los vértices que son adyacentes entre sí, manteniendo la diagonal principal con elementos nulos (figura 3). Este tipo de representación se encuentra definida por:

$$(A)_{ij} = \begin{cases} 1, & \text{si } v_i v_j \in E(G) \\ 0, & \text{de otro modo} \end{cases}$$

Ejemplo:



a)

A =

0	1	1	1	0	0
1	0	1	0	1	0
1	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	1
0	0	1	0	1	0

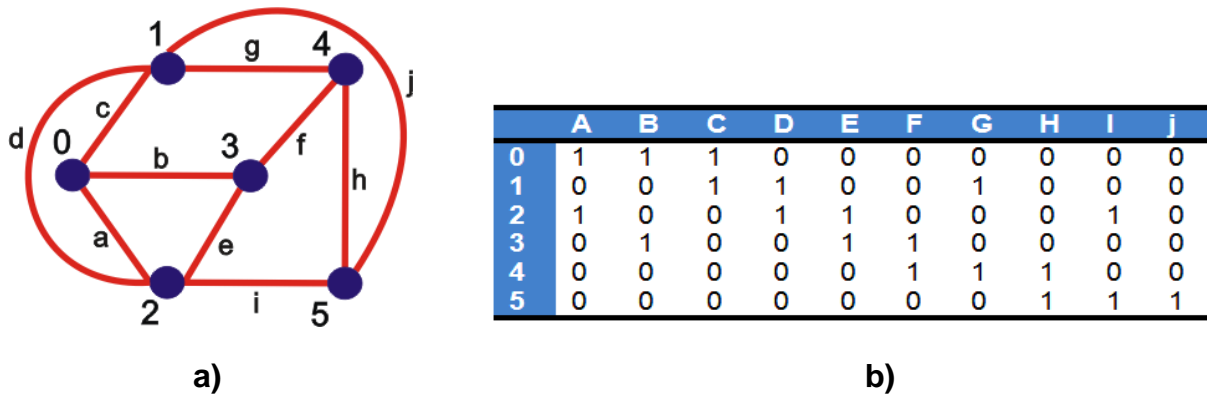
b)

**Fig. 3.** a) Grafo disyuntivo, b) Matriz de Adyacencia del grafo a)

- *Matriz de Incidencia:* Es una matriz  $n \times m$  que permite representar el número de aristas incidentes en un vértice (figura 4), de modo que se encuentra definida por:

$$(B)_{ij} = \begin{cases} 1, & \text{si } v_i \text{ es incidente con } e_j \\ 0, & \text{de otro modo} \end{cases}$$

Ejemplo:



**Fig. 4.** a) Grafo disyuntivo, b) Matriz de Incidencia del grafo a)

### III. Máquinas en Paralelo No Relacionadas

El problema de Máquinas en Paralelo no Relacionadas (UPMP), por sus siglas en inglés (*Unrelated Parallel Machines Problem*) es una variante del problema clásico de Calendarización de Trabajos en Talleres de Manufactura, a partir del cual se puede obtener un mapeo al UPMP [12] [13].

De acuerdo a la naturaleza del problema, ha sido clasificado como problema NP [5], debido a que conserva características de su antecesor, teniendo como diferencia esencial que al ser máquinas no relacionadas, las máquinas tienen capacidades diferentes, por lo que el tiempo de procesamiento para cada operación (costo) dependerá de la máquina así como de la posición a la que ésta sea asignada.

El problema puede ser descrito de forma general como el conjunto de  $n$  trabajos independientes que requieren ser calendarizados en  $k$  posiciones correspondientes a  $m$  máquinas en paralelo no relacionadas, de forma que se cumpla la función objetivo del problema; ya sea minimizar o maximizar; para ello se debe considerar que un trabajo no puede ser procesado más de una vez, además de que los trabajos no pueden ser interrumpidos una vez que se han asignado [4]. Para realizar la calendarización de los trabajos es necesario tomar en cuenta las siguientes características propias del problema, de las cuales algunas son tomadas de forma implícita:

1. Un trabajo debe ser procesado únicamente una vez.
2. No existen restricciones o precedencias entre trabajos diferentes.

3. Un trabajo puede ser procesado en cualquier máquina.
4. Cada posición puede ser ocupada a lo más por un trabajo (Restricción de Capacidad de Recursos).
5. El tiempo de procesamiento de cada trabajo varía de acuerdo a la máquina y posición al que sea asignado.
6. No se consideran interrupciones (*preemptions*) ni tiempos de preparación (*Setup Times*).

El tiempo en el que todos los trabajos han sido calendarizados, es mejor conocido como el *tiempo total de término de todos los trabajos*.

Tomando en cuenta esta definición conceptual, el problema de UPMP puede ser definido de manera formal como el conjunto de trabajos  $J = \{1, 2, \dots, n\}$  que necesitan ser calendarizados en las posiciones  $K = \{1, 2, \dots, n\}$ , correspondientes a cada una de las máquinas del conjunto  $I = \{1, 2, \dots, m\}$ ; donde no se permiten interrupciones y el tiempo de procesamiento  $P$  de cada trabajo  $j$ , depende de la máquina a la que el trabajo sea asignado, teniendo como función objetivo encontrar la calendarización que minimice el tiempo total de término de todos los trabajos.

Las características antes mencionadas encajan de manera adecuada con los requerimientos de los sistemas de manufactura utilizados actualmente en la industria, ya que mientras la demanda se incrementa, los requerimientos en cuanto a maquinaria se refiere se hacen mayores, lo que hace indispensable la adquisición de equipo nuevo; ésta es la razón principal por lo que las capacidades de las máquinas son diferentes; esta característica es la que define al problema, ya que se toman en cuenta las capacidades de las máquinas para llevar a cabo una calendarización eficiente y de esta forma evitar el gasto excesivo e innecesario de adquisición de maquinaria y equipo.

El problema de Máquinas en Paralelo no Relacionadas, puede ser formulado por medio de Programación Lineal Entera Binaria, lo cual permite tratar al problema por medio de métodos no determinísticos, mejor conocidos como heurísticas [1], debido a su complejidad. El modelo de programación lineal entera binaria presenta al problema como un conjunto  $n$  de trabajos que deben ser procesados en un conjunto  $m$  de máquinas, las cuales cuentan con  $k$  posiciones cada una. Cada trabajo puede ser procesado en cualquier máquina y cada máquina puede procesar cualquier trabajo, de modo que el tiempo de procesamiento de cada trabajo dependerá de la máquina y la posición en la que éste sea ejecutado. Estas restricciones son tomadas en cuenta para llevar a cabo el modelado gráfico del problema, debido a que son las características básicas que definen al problema. Dichas restricciones se muestran en la formulación matemática y se enlistan a continuación.

- (2) Cada trabajo  $j$  puede ser procesado solo una vez.
- (3) Cada posición  $k$  de cada máquina  $i$ , puede procesar a lo más un trabajo.
- (4) Indica si un trabajo  $j$  fue asignado ó no a la posición  $k$  de una máquina  $i$ .

A continuación, en la figura 5 se muestra la formulación matemática del UPMP analizada para llevar a cabo su representación gráfica.

$$\min f = \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^n kP_{ij} X_{ikj} \quad (1)$$

Sujeto a:

$$\sum_{i=1}^m \sum_{k=1}^n X_{ikj} = 1 \quad j = 1, \dots, n \quad (2)$$

$$\sum_{j=1}^n X_{ikj} \leq 1 \quad i = 1, \dots, m, k = 1, \dots, n \quad (3)$$

$$X_{ikj} \in \{0,1\} \quad i = 1, \dots, m, k = 1, \dots, n, j = 1, \dots, n \quad (4)$$

**Fig. 5.** Modelo de Programación Lineal Entera Binaria para el problema de Máquinas en Paralelo no Relacionadas [4].

#### IV. Colonia de Hormigas

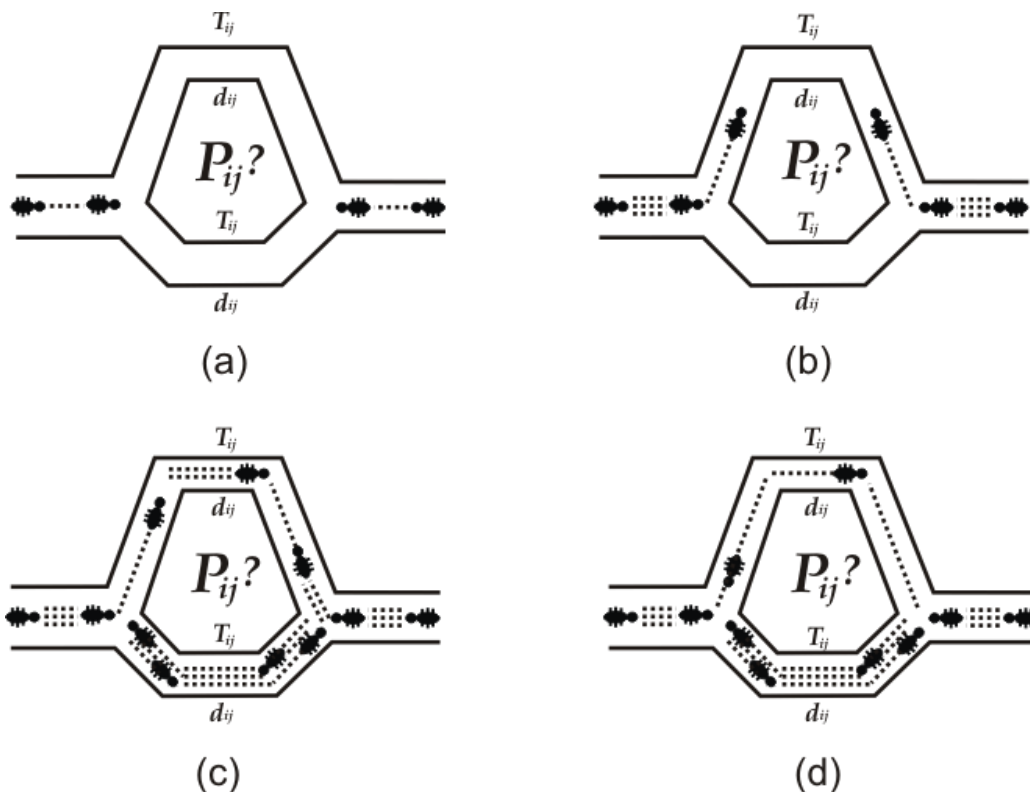
Colonia de Hormigas es una metaheurística inspirada en el comportamiento inteligente y estructurado de las hormigas reales, la cual fue propuesta por Marco Dorigo en 1991 y fue aplicada por primera vez al problema del Agente Viajero [6, 7, 8, 9, 10]. Métodos bioinspirados como Colonia de Hormigas han motivado a muchos investigadores a tratar de entender la interacción que existe entre las hormigas para construir ordenadas comunidades, además de llevar a cabo una especie de comunicación indirecta, denominada *Stigmergy* [11], la cual describe una comunicación entre los individuos de una colonia por medio de elementos o sustancias químicas.

La metaheurística hace una analogía con el comportamiento de las hormigas reales, de manera general, las hormigas artificiales van explorando de forma iterativa diversos caminos, hasta llegar a la fuente de alimento; mientras cada hormiga se encuentra moviéndose de un lugar a otro, cada una va depositando a su paso una sustancia química denominada *feromona*. Este rastro hace que el camino sea más atractivo a las demás hormigas, es decir, tendrá mayor probabilidad de que las demás hormigas se vean atraídas hacia los caminos con mayor monto de feromona, así mismo se lleva a cabo una evaporación natural de esta sustancia de acuerdo al tiempo y la distancia, por lo que al final, el camino más corto será el que contenga mayor intensidad de feromona y por ende el más transitado.

Al inicio del procedimiento, se tiene una distribución de las hormigas en cada uno de los nodos para iniciar el proceso de construcción de soluciones. Cada una de las hormigas cuenta con una lista tabú, la cual funciona como la memoria de la hormiga, es decir, va almacenando los nodos recorridos de modo que impide que un nodo sea visitado más de una vez por una misma hormiga, esto se realiza con la finalidad de asegurar la factibilidad de las soluciones construidas acorde con las restricciones especificadas en la formulación matemática del problema.

Mientras cada hormiga va realizando su recorrido, va dejando a su paso un rastro de feromona, de modo que mientras más transitado sea un camino, mayor será el monto de feromona y por ende se evitará su rápida evaporación. En este proceso, la feromona tiene un papel fundamental, debido a que al llevar a cabo la evaporación se genera una especie de olvido para las hormigas, con lo que se evita la convergencia prematura del algoritmo y se permite la exploración de nuevos caminos. De modo que, cuando una hormiga encuentra un buen recorrido, este tendrá mayor probabilidad de ser elegido por otras hormigas, por lo que el rastro de feromona se va reforzando en los caminos más cortos, haciendo finalmente que el mejor recorrido sea encontrado. A continuación, en la figura 6 se muestra un ejemplo del recorrido que realizan las hormigas artificiales para la exploración y búsqueda de soluciones.





**Fig. 6.** Ejemplo del recorrido realizado por las hormigas, donde se evalúa la probabilidad de transición de un nodo a otro [9].

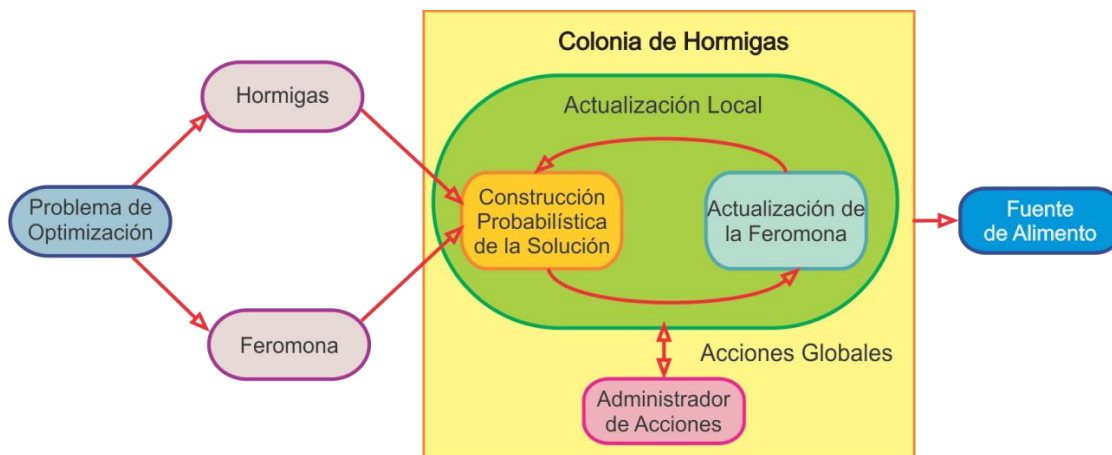
De acuerdo a la figura 6, en el inciso (a) cuando  $t = 0$  (iteración = 0), no existe aún un rastro de feromona que puedan seguir las hormigas artificiales, de modo que comienzan con una exploración realizada aleatoriamente, conforme el proceso de construcción de soluciones va avanzando, (b), (c) y (d), los montos de feromona van cambiando, lo que hace que la hormiga pueda realizar una evaluación de los caminos y tomar decisiones probabilísticas de acuerdo a los criterios establecidos en la formulación matemática y a las características propias del método de optimización.

Para tener una idea más clara del procedimiento explicado anteriormente en la figura 6 correspondiente al proceso de construcción de soluciones, se presenta en la figura 7 un diagrama de funcionamiento que muestra de forma esquemática los procesos que se efectúan durante el método Colonia de Hormigas Clásico, donde al principio se inicializan las hormigas y los montos de feromona, posteriormente, en cada iteración, cada hormiga va llevando a cabo el proceso de construcción de soluciones de manera probabilística, hasta que cada hormiga construye una solución de acuerdo a las características y restricciones propias del problema, de modo que durante cada iteración se lleva a cabo:

- *Incremento de feromona:* Cada hormiga incrementa la feromona en los arcos almacenados en su lista tabú. Actualización local de la Feromona.

- *Evaporación*: Proceso global que se realiza en todos los arcos, independientemente de si se han recorrido o no.
- *Actualización de la feromona*: Se realiza únicamente en los arcos correspondientes a la mejor solución hasta el momento.

Cabe mencionar que las dos últimas acciones son llevadas a cabo por un proceso denominado en este trabajo de investigación *Administrador de Acciones*. Todo el procedimiento explicado anteriormente, se realiza de forma iterativa hasta que el mejor camino a la fuente de alimento sea encontrado.



**Fig. 7.** Diagrama esquemático de Colonia de Hormigas.

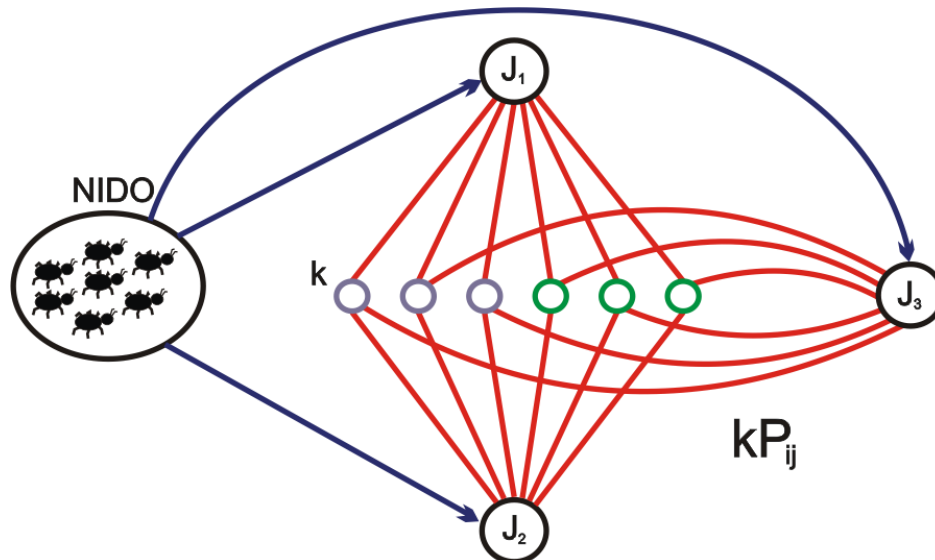
## V. Representación del UPMP para Colonia de Hormigas

La representación de un problema por medio de grafos se lleva a cabo para modelar las propiedades combinatorias de un problema, con la finalidad de facilitar su entendimiento, por lo que es de vital importancia que al desarrollar un grafo, este cumpla con ciertas características, las cuales se especifican en la literatura [15, 14].

- *Reglas Básicas*. Se refieren a aspectos elementales, como el solapamiento entre aristas, vértices o ambos.
- *Reglas Semánticas*. Son reglas de posicionamiento de vértices y del trazo de arcos o aristas (enrutado).
- *Reglas Estructurales*. Son reglas de posicionamiento y enrutado especificadas por la teoría de grafos. Por ejemplo, colocar los vértices de mayor orden en el centro del grafo, minimizar la longitud total de las aristas o minimizar el número de cruces entre vértices.

Para llevar a cabo un buen modelado del problema de Máquinas en Paralelo no Relacionadas y tomando en cuenta las reglas mencionadas anteriormente, se propone una representación del problema para Colonia de Hormigas mediante la utilización de un grafo disyuntivo, para el cual se analizaron las características básicas del problema, de las cuales se tomaron solo aquellas que se expresan de forma explícita en el modelo matemático, incorporando las características de Colonia de Hormigas.

De acuerdo al problema tratado, se tiene que cada hormiga elige el siguiente trabajo a calendarizar de forma aleatoria, debido a que no existen dependencias o restricciones de precedencia entre trabajos diferentes. Una vez que un trabajo ha sido elegido, se procede a llevar a cabo el proceso de calendarización, para el cual cada hormiga analiza las diferentes opciones calculando la probabilidad de transición, es decir, calcula la probabilidad de acuerdo a la distancia heurística, tomando aquella opción que conlleve la mayor probabilidad. A continuación (Figura 8) se muestra la representación explicada anteriormente para una instancia de 3 trabajos y 2 máquinas.



**Fig. 8.** Representación del UPMP para Colonia de Hormigas por medio de un grafo disyuntivo [16].

En esta representación se muestra que la elección de cada uno de los trabajos, se lleva a cabo de forma aleatoria sin ninguna ponderación, a diferencia del proceso de calendarización, donde para seleccionar una posición de una máquina, ésta tiene que ser evaluada de forma probabilística de acuerdo a las restricciones propias del problema y basándose en las características del método de solución.

Cabe mencionar que todas las hormigas tienen acceso a una *matriz de costos*, la cual contiene la distancia heurística existente entre cada uno de los vértices del grafo, correspondiente al tiempo de procesamiento de cada uno de los trabajos en cada una de las posiciones de las diferentes máquinas. Otro punto importante es que el número

de posiciones en cada máquina está en función del número de trabajos, además de que los costos dependen directamente de la posición y máquina al que el trabajo sea asignado, debido a que cada posición dentro de una misma máquina cuenta con una penalización especificada en la formulación matemática, de modo que la primera posición de una máquina será la que conlleve el menor costo, el cual se irá incrementando conforme avanza el número de posiciones dentro de una misma máquina. En la figura 9 se muestra un ejemplo de la representación matricial para una instancia de 3 trabajos y 2 máquinas.

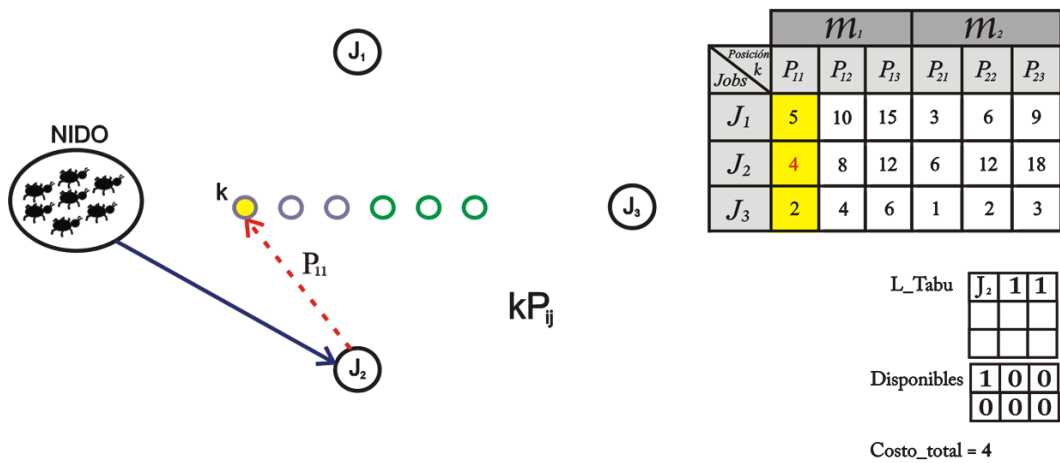
		$M_1$			$M_2$		
Jobs	Posición $k$	$P_{11}$	$P_{12}$	$P_{13}$	$P_{21}$	$P_{22}$	$P_{23}$
	$J_1$		4	8	12	7	14
$J_2$		2	4	6	5	10	15
$J_3$		1	2	3	6	12	18

**Fig. 9.** Representación matricial de costos para el UPMP [16]

La factibilidad de las soluciones obtenidas se evalúa de acuerdo a la información de la estructura de datos (Lista Tabú, Matriz de posiciones disponibles y Costo del recorrido) con la que cuenta cada hormiga, la cual asegura que los recorridos de las hormigas no violen ninguna de las restricciones especificadas en la formulación matemática del problema, por lo que los pasos realizados por una hormiga para construir una solución a una instancia de 3 trabajos y 2 máquinas de acuerdo al grafo propuesto, son ejemplificados en las figuras 10 a la 12.

### **Paso 1:**

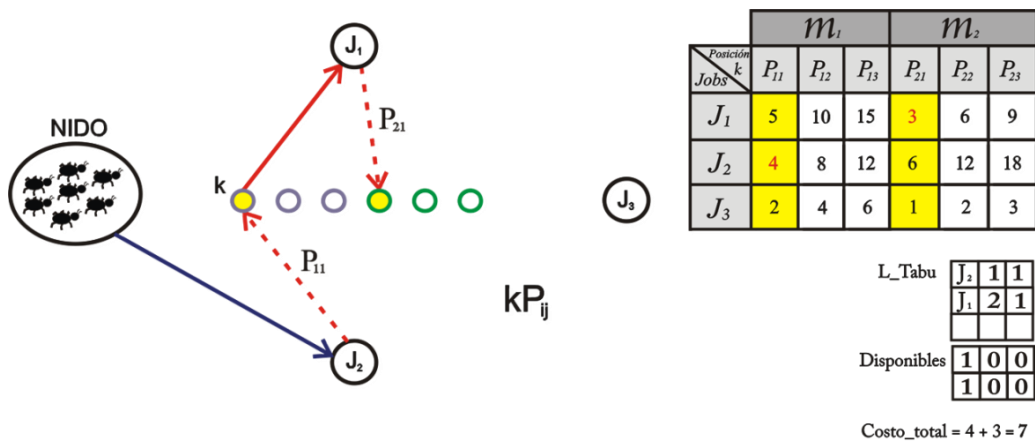
Al inicio del recorrido, la hormiga elige un trabajo de forma aleatoria y lo agrega a su *lista tabú*, cambiando el estado de dicho trabajo en la *matriz de posiciones disponibles* (0 - no ha sido calendarizado ó 1 - si ya se ha procesado dicho trabajo) y de la *matriz de costos* elige aquella posición que conlleve el menor costo, de acuerdo al trabajo elegido, guarda la posición y la máquina utilizada dentro de la lista tabú, marcando la posición elegida para que no vuelva a ser utilizada por otro trabajo, lo cual es una restricción del problema, mientras va almacenando el costo del recorrido en la variable definida *Costo\_total*.



**Fig. 10.** Paso 1. Recorrido de una hormiga para la construcción de una solución en un grafo disyuntivo de UPMP para Colonia de Hormigas [16].

**Paso 2:**

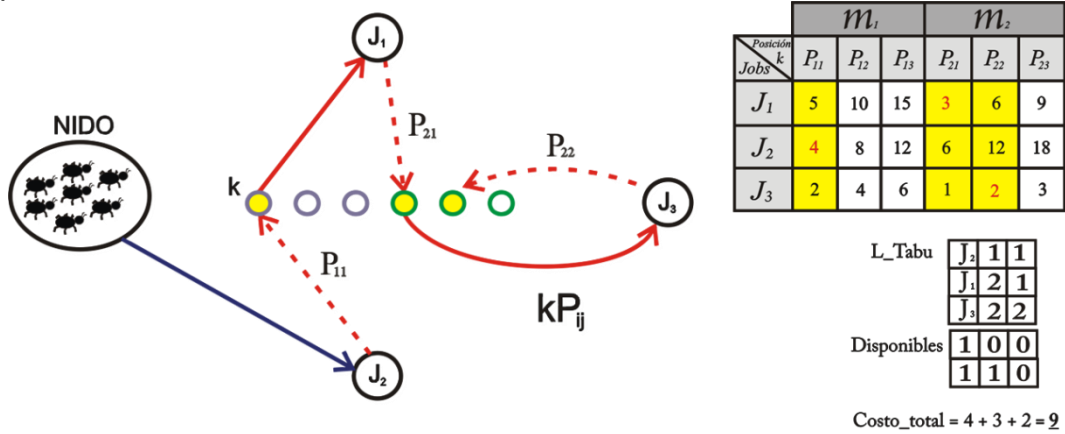
La hormiga vuelve a elegir un trabajo de forma aleatoria y verifica en su *lista tabú* que dicho trabajo no se haya calendarizado previamente, de lo contrario elige otro trabajo y repite el procedimiento, si este trabajo no se ha calendarizado, lo agrega a su *lista tabú* y cambia el estado del trabajo en la *matriz de posiciones disponibles*, en la *matriz de costos* toma aquella posición que conlleve el menor costo para el trabajo seleccionado y que no haya sido utilizada previamente, siendo así, guarda la información en su lista tabú y marca la posición al igual que en el paso anterior, en consiguiente incrementa su *Costo\_total* de acuerdo al costo de la posición seleccionada.



**Fig. 11.** Paso 2. Recorrido de una hormiga para la construcción de una solución en un grafo disyuntivo de UPMP para Colonia de Hormigas [16].

**Paso 3:**

En este paso, la hormiga vuelve a realizar el mismo procedimiento de selección llevado a cabo en el paso 2, de modo que una vez que la lista tabú se ha completado, la hormiga obtendrá una calendarización factible de trabajos, así como el costo total correspondiente al recorrido total realizado.



**Fig. 12.** Paso 3. La hormiga completó un recorrido en el grafo disyuntivo de UPMP para Colonia de Hormigas y se tiene el Costo total del recorrido [16].

De acuerdo a la solución particular mostrada en las figuras 10 -12, se puede observar claramente que la estructura de datos manejada por cada una de las hormigas tiene una función primordial, ya que sirve como punto de referencia para comprobar las restricciones del problema, por lo que al final de la calendarización, ésta contendrá la información total del recorrido realizado por la hormiga garantizando su factibilidad.

## VI. Conclusiones

Los grafos son estructuras combinatorias de gran importancia dentro de las Ciencias Computacionales, debido a que permiten modelar problemas de diversa índole, facilitando su entendimiento.

Aunado a la importancia del uso de grafos en diversas áreas, el desarrollo de un grafo que modele un problema de gran relevancia como el problema de Máquinas en Paralelo no Relacionadas para un método de solución como lo es Colonia de Hormigas, es una aportación que permite comprender y visualizar de mejor manera el comportamiento del método propuesto para un problema dado, lo que facilita el proceso de desarrollo computacional. Para ello fue necesario tomar en cuenta reglas de diseño de grafos especificadas en la literatura, ya que es tanta su importancia, que un mal modelado puede traer como consecuencia una incorrecta comprensión del problema en general, lo que conlleva a complicaciones en el desarrollo computacional.

De acuerdo al análisis realizado al grafo propuesto, se concluye que el modelado cumple con el objetivo planteado, ya que permite obtener una mejor comprensión del funcionamiento de Colonia de Hormigas aplicado al UPMP, pudiendo compararlo manualmente al realizar el recorrido de una hormiga para una solución particular.

## Referencias

- [1]. Papadimitrou Christos H., Steiglitz Kenneth. Combinatorial Optimization. Algorithms and Complexity. ISBN. 0-486-40258-4, Dover Publications, Inc. Mineola, New York, 1998.
- [2]. Comellas Francesc, Fàbrega Josep, Sànchez Anna, Serra Oriol. Matemática Discreta. ISBN. 84- 8301-456-4. Universidad Politécnica de Cataluña, S. L. Ediciones UPC, 2001.
- [3]. Diestel Reinhard. Graph Theory. Graduate Texts in Mathematics. Third Edition. ISBN. 0072- 5285, ISBN-10 3-540-26183-4, ISBN-13 978-3-540-26183-4. Springer-Verlag Berlin Heidelberg. Alemania, 2006.
- [4]. Pinedo Michael L. Scheduling Theory, Algorithms, and Systems. Third Edition. New York University. ISBN: 978-0-387-78934-7, e-ISBN: 978-0-387-78935-4. Ed. Prentice Hall. Springer, 2008.
- [5]. Garey M. R., D. Johnson. “Computers and Intractability” a Guide to the Theory of NP-Completeness. Freeman. New York NY. ISBN 0-7167-1044-7. 1979.
- [6]. Dorigo M., Maniezzo V., Colorni A. Positive Feedback as a Search Strategy. Technical report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1991a.
- [7]. Dorigo M., Maniezzo V., Colorni A. The Ant System: An Autocatalytic Optimizing Process. Technical Report 91-016 Revised, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1991b.
- [8]. Dorigo M. Optimization, Learning and Natural Algorithms [in Italian]. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1992.
- [9]. Dorigo M., Maniezzo V., Colorni A. Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics–Part B, 26(1):29– 41, 1996.

- [10]. Dorigo M., Stützle T. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. Handbook of Metaheuristics, Springer. 2003.
- [11]. Grassé P. P. La Reconstruction Du Nid et Les Coordinations Inter-Individuelles Chez *Bellicositermes Natalensis* et *Cubitermes Sp.* La théorie de la Stigmergie: Essai d'Interpretation Du Comportement Des Termites Constructeurs, pages 41-81, 1959.
- [12]. Cruz-Chávez Marco Antonio, Martínez-Oropeza Alina, Zavala-Díaz José Crispín, Martínez-Rangel Martín G. Relajación del Problema de Calendarización de Trabajos en un Taller de Manufactura utilizando un Grafo Bipartita. 7mo. Congreso Internacional de Cómputo en Optimización y Software AGECOMP-CICos 2009. ISBN: 978-970.9750-26.3. 2009
- [13]. Cruz-Chávez Marco Antonio, Martínez Oropeza Alina, Rivera López Rafael. Relaxation of Job Shop Scheduling Problem using a Bipartite Graph. Presentado como Ponencia en Cerma 2010. To publish in IEEE. 2010.
- [14]. Kozo Sugiyama. Graph Drawing and Applications for Software and Knowledge Engineers. ISBN. 981-02-4879-2. Vol. 11. World Scientific. 2002.
- [15]. Di Battista Giuseppe, Eades Peter, Tamassia Roberto , Tollis Ioannis G. Graph Drawing. Algorithms for the Visualization of Graphs. ISBN. 0-13-301615-3. Ed. Prentice Hall. USA, 1999.
- [16]. Martínez Oropeza Alina. Solución al Problema de Máquinas en Paralelo No Relacionadas Mediante un Algoritmo de Colonia de Hormigas. Tesis Profesional para obtener el grado de Maestro en Ingeniería y Ciencias Aplicadas. CIICAP-UAEM. Cuernavaca, Mor., México, 2010