

PROGRAMACIÓN DE PROYECTOS DE LA EMPRESA GEOFUNDACIONES S.A. MEDIANTE LA ASIGNACIÓN DE MÁQUINAS UNIFORMES EN PARALELO PARA LA REDUCCIÓN DE LA TARDANZA PONDERADA TOTAL

Orlando Federico González Casallas, M.Sc. Ingeniería Industrial.

Universidad Pontificia Bolivariana Seccional Bucaramanga

Recibido octubre 1, 2009-Aceptado octubre 27, 2009

<http://dx.doi.org/10.18566/puente.v3n2.a02>

Resumen—Este artículo se centra en la solución de la programación de máquinas uniformes en paralelo con el propósito de minimizar la tardanza ponderada total en la consecución de los proyectos de construcción, a través de la metaheurística *Discrete Particle Swarm Optimization (DPSO)*. Inicialmente, se procede con la caracterización del problema mediante los conceptos teóricos de la programación de producción; seguidamente se propone una heurística para encontrar la solución inicial del problema de asignación de máquinas en los proyectos de construcción, la cuál se mejora mediante la metaheurística *DPSO*. Se presenta un método de reprogramación de máquinas en el caso que se presente la llegada de nuevos proyectos de construcción; por último, se considera la calidad de solución del algoritmo mediante la comparación de soluciones óptimas en problemas clásicos de la literatura, obtenidas a través de programación lineal entera mixta.

Palabras clave— Programación de Producción, Máquinas uniformes, Tardanza ponderada total, *Discrete Particle Swarm Optimization*, Reprogramación.

Abstract— This paper focus in parallel uniform parallel machines scheduling solution to minimize the total weighted tardiness in finish construction projects through the *Discrete Particle Swarm Optimization (DPSO)* metaheuristic. First, the characterization of the problem through the concepts of scheduling theory is showed; after that, the heuristic to find an initial solution to assignment machines in construction projects is proposed. Rescheduling machines method in case that arrive new construction projects is showed; finally, the algorithm solution quality through the comparison with optimal solution found with mixed integer linear programming is considered.

Keywords— Scheduling, Uniform Machines, Total Weighted Tardiness, *Discrete Particle Swarm Optimization*, Rescheduling.

Artículo basado en la Tesis de Maestría en Ingeniería Industrial con énfasis en Sistemas de Producción y Logística de la Universidad de los Andes.

O. González Ingeniero Industrial de la Universidad Libre de Colombia (Seccional Cúcuta) 2006. M.Sc. en Ingeniería Industrial con énfasis en sistemas de producción y logística de la Universidad de los Andes 2009. Teléfono 320-26248962. e-mail: orlando.gonzalez@upbga.edu.co.

I. INTRODUCCIÓN

LOS sistemas de construcción que consideran la utilización de varias máquinas procesadoras de trabajos, se pueden encontrar en sectores particulares de la industria de la construcción; por lo tanto, es importante controlar los tiempos requeridos para la construcción de obras para cumplir con lo acordado en las licitaciones de construcción.

El presente trabajo está enmarcado en el desarrollo de un aplicativo para la programación de máquinas en la consecución de proyectos de construcción, basado en técnicas de la programación de producción. Este enfoque se realiza con el fin de obtener una herramienta que permita tomar decisiones con respecto a la asignación de máquinas a los proyectos y a su vez, identificar problemáticas referentes al uso de máquinas o incumplimiento de las fechas de entrega de los proyectos, teniendo en cuenta la utilidad monetaria esperada de los proyectos de construcción en un periodo determinado de tiempo.

Adicionalmente, en las empresas de construcción es necesario el esquema de la reprogramación de proyectos de construcción, puesto que se debe disponer de la misma maquinaria para el desarrollo de dichos proyectos; con esto en mente, el esquema de reprogramación de producción será tenido en cuenta en el desarrollo de este trabajo.

El contenido de este documento está organizado de la siguiente forma: La sección II, se considera la descripción general de la problemática de la empresa y definiciones importantes dentro de la terminología de los procesos de construcción. En la sección III, se encuentra el marco teórico, revisión bibliográfica sobre el método propuesto de solución al

problema de la empresa y diversos enfoques de solución referentes a sistemas de producción con máquinas en paralelo. En la sección IV, se realiza la descripción de la propuesta de solución basada en el algoritmo de Optimización de Enjambre de Partículas Discreto “*Discrete Particle Swarm Optimization*” (DPSO). La sección V, presenta los resultados computacionales obtenidos para los problemas clásicos de máquinas uniformes en paralelo para reducir el tiempo de terminación de los trabajos y la tardanza total, considerando la comparación de dichos resultados con la solución obtenida con el algoritmo propuesto en este trabajo; finalmente en la sección VI, se presenta las conclusiones obtenidas en esta investigación.

II. DESCRIPCIÓN DE LA PROBLEMÁTICA DE LA EMPRESA

La empresa considerada para este caso de estudio, se dedica a la realización de obras de construcción, específicamente a la solución de problemas relacionados con la geotecnia aplicada, tanto en obras de desarrollo urbano como de infraestructura. Geofundaciones S.A., dispone de maquinaria pesada para obras de cimentación, contención de tierra, pequeña perforación y tablestacado, tal como se presenta en la Tabla I.

TABLA I
MAQUINARIA QUE DISPONE LA EMPRESA PARA LOS PROYECTOS DE CONSTRUCCIÓN

Obras	Tipo de Máquinas
Cimentación	Pilotes tipo Kelly Pilotes tipo Hélice
Contención de tierra	Pantallas y Barretes
Tablestacado	Equipo de tablestacado
Pequeña Perforación	Anclajes Micropilotes

Fuente: Autor.

Cimentación: Es la parte de una estructura que se encarga de transmitir cargas que soporta una edificación en particular a la superficie. Para este fin, la empresa dispone de máquinas conocidas como pilotes que se encargan de realizar

perforaciones profundas en el suelo para sentar la base de la cimentación.

Contención de Tierra: Es la obra encargada de hacer uso de áreas bajo tierra para la construcción de plantas de sótano en obras de edificación. La empresa actualmente cuenta con maquinaria para la realización de pantallas en concreto reforzado.

Pequeña perforación: Son obras especializadas en el refuerzo de estructuras. La empresa cuenta con anclajes y micropilotes para llevar a cabo este fin.

Tablestacado: Son obras utilizadas para evitar la erosión y penetración de agua al terreno de la construcción. La empresa cuenta con equipos que reciben el mismo nombre.

Actualmente, la empresa realiza la programación de los equipos (máquinas) de obras de construcción en forma manual, la cual dura aproximadamente una semana. La programación resultante es enviada a los ingenieros de construcción para realizar la planificación de operarios y demás recursos que son necesarios en la consecución de los proyectos.

La importancia de esta investigación, radica en ofrecer una herramienta de apoyo para el cumplimiento de los proyectos de construcción que le generen una alta utilidad esperada; e indirectamente terminar a cabalidad los proyectos según lo pactado en la licitación, pues la tardanza en la terminación de algún proyecto puede encausar en incrementar costos de operaciones con respecto a la utilización de equipos de apoyo y de recurso humano. Por ende, se considera como objetivo fundamental del modelo, determinar la importancia de terminación de los proyectos de construcción en función de la utilidad esperada.

Un proyecto de construcción en particular está constituido por una serie de obras, las cuales deben completarse durante el tiempo especificado en la licitación firmada por la empresa y el cliente. Por ejemplo, suponer que un proyecto de la empresa Geofundaciones S.A., consiste en la realización de una obra de cimentación y contención de tierra. Una posible representación de la asignación de obras de construcción a los proyectos, se presenta en la Figura 1.

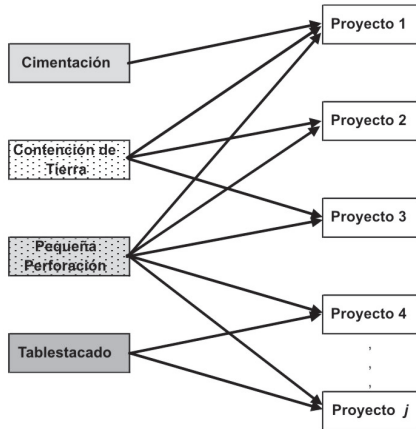


Figura 1. Representación de una posible asignación de obras a los proyectos de construcción

III. MARCO TEÓRICO

Los sistemas actuales de programación de producción se basan en el conocimiento de información sobre los trabajos y las máquinas para definir estrategias que permitan alcanzar uno o varios objetivos. Según Ramamritham y Stankovic, “*la nueva generación de sistemas de programación de producción permitirán diseñar estrategias de solución dinámicas, predictibles y flexibles*” [1]. En el caso de los sistemas de producción en el sector de la construcción, es importante el uso de herramientas computacionales que presenten resultados dinámicos y flexibles para lograr una planificación y toma de decisión adecuada en el uso de recursos y maquinaria, necesaria en la consecución de los proyectos.

Hoy en día, los sistemas de producción en el sector de la construcción consideran el uso de múltiples máquinas que permitan la terminación de proyectos en un menor tiempo. Según la forma como es utilizado un grupo de máquinas, el ambiente de producción se puede clasificar en máquinas en paralelo, cuando un grupo de máquinas es asignado para procesar un trabajo en particular al mismo tiempo; mientras que si un grupo de máquinas especializadas en varios trabajos es asignado para procesar un tipo particular de trabajo al mismo tiempo, se conoce

como un sistema de producción multiprocesador de trabajos.

Según la problemática descrita anteriormente, el problema se basa en la asignación de máquinas especializadas a cada tipo de obra que constituyen un proyecto en particular; por lo tanto, la asignación de máquinas a los proyectos se puede abordar mediante técnicas propias de la programación de producción. Suponer que se dispone de un conjunto N de proyectos $N = \{J_1, J_2, \dots, J_N\}$, los cuales deben ser procesados en un conjunto m de máquinas especializadas para el tipo de obra k $m_k = \{m_1, m_2, \dots, m_{k_j}\}$. Un grupo de máquinas m_k en particular, puede procesar un tipo de obra k requerida en un proyecto, teniendo en cuenta la velocidad de procesamiento V_i .

Cada proyecto J_j está constituido por una secuencia de obras $J_j = \{O_{j1}, O_{j2}, \dots, O_{jk_j}\}$, y por un tiempo de procesamiento p_j , que es independiente de las M máquinas disponibles. Una vez iniciada una obra en particular, esta debe completarse sin interrupción.

La siguiente notación será usada en la propuesta de solución del problema:

N	Cantidad de proyectos
i	Cantidad de máquinas
k	Tipo de obra ($k=1, 2, \dots$)
j	Cantidad de proyectos
J_j	Proyecto j
m_k	Número de máquinas disponibles para el tipo de obra k
$side_{jk}$	Máximo número de máquinas requeridas para procesar la obra k del proyecto j
p_{ikj}	Tiempo de procesamiento de la obra k del proyecto j en la máquina i
p_j	Tiempo de procesamiento del proyecto j
v_i	Velocidad de la máquina i
r_j	Tiempo de inicio del proyecto j
d_j	Fecha de entrega del proyecto j
T_j	Tardanza del proyecto j (medido en días)
u_j	Utilidad esperada del proyecto j (medido en millones de pesos)
w_j	Ponderación basada en la utilidad esperada de la terminación del proyecto j

El problema de asignación de máquinas para los proyectos de construcción, se puede entender a través de la teoría de programación de producción como la programación de máquinas uniformes; es decir, máquinas en paralelo con diferente velocidad de procesamiento, para minimizar la tardanza ponderada total de los proyectos; ya que según lo descrito en la sección anterior, la empresa debe terminar los proyectos de construcción según lo pactado con los clientes, teniendo en cuenta la utilidad esperada por la terminación de los proyectos; y de acuerdo a la restricción de elegibilidad de máquinas especializadas en el procesamiento de una obra que requiere un proyecto en particular y el tiempo de inicio para cada proyecto.

Considerar el siguiente ejemplo; la empresa tiene a cargo cuatro proyectos de construcción los cuales tienen una estimación de tiempo de procesamiento (en semanas) p_j . Para este fin, se dispone la siguiente información que se presenta en la Tabla II.

TABLA II
MAQUINARIA QUE DISPONE LA EMPRESA PARA LOS PROYECTOS DE CONSTRUCCIÓN

Proyecto	Cimentación (1)	Contención Tierra (2)	Pequeña Perforación (3)	Tiempo (p_i)
1	X		X	5
2	X	X		4
3		X	X	6
4	X	X		7

Fuente: Autor

TABLA III
MAQUINARIA QUE DISPONE LA EMPRESA PARA LOS PROYECTOS DE CONSTRUCCIÓN

Cimentación	Contención de Tierra	Pequeña Perforación
TH-15	LB14	CMV-1
B125-1	LB15	CMV-2
B125-2	LS98	CMV-3

Fuente: Autor

La Tabla 3 presenta la información de los equipos que se disponen para la consecución de las obras. Suponer que los i equipos de cimentación, contención de tierra y pequeña perforación tienen una velocidad de procesamiento de

$V_{i1} = 3$, $V_{i2} = 4$ y $V_{i3} = 2$, respectivamente. Una obra en particular puede disponer de varias máquinas para su procesamiento; adicionalmente no existen precedencias entre las obras de un proyecto en particular, puesto que se desea aprovechar al máximo los recursos disponibles para la terminación de un proyecto. Con base en lo expuesto anteriormente, una posible solución para el ejemplo planteado se presenta en la Figura 2.

Una representación adecuada para el problema se basa en la notación para problemas teóricos de programación de producción propuesta por Graham [2], Veltman [3] y Drozdowski [4]. Sea Qm_k el ambiente de producción consistente en un grupo de máquinas uniformes disponibles para procesar la obra k ; m_k representa la restricción de elegibilidad de máquinas para procesar la obra k ; r_j , representa el tiempo de inicio del proyecto j . La función objetivo definida como la tardanza ponderada total de los proyectos, es representada por $\sum w_j T_j$. Donde el factor de ponderación se define como la relación entre la utilidad esperada del proyecto j , y la utilidad esperada total $w_j = u_j / \sum u_j$. Por lo tanto, el problema se puede representar por la siguiente tripleta: $Qm_k / m_k, r_j / \sum w_j T_j$.

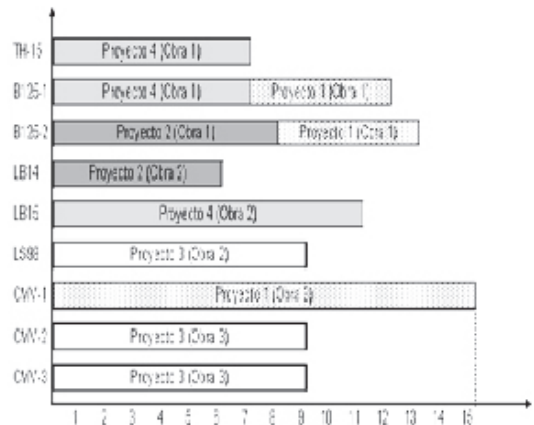


Figura 2. Posible solución para el problema de asignación de máquinas a los proyectos de construcción. Tiempo en días

En este modelo varias máquinas especializadas pueden realizar una obra en particular al mismo tiempo y es permitido que varias obras se

procesen al mismo tiempo; puesto que se debe tener en cuenta la utilización de recursos disponibles para la realización de la obra, como por ejemplo la materia prima y la mano de obra. Considerar el siguiente ejemplo, suponga que un proyecto consta de dos obras. Se dispone de un grupo de máquinas uniformes para cada obra, definido de la siguiente forma: $m_{obra1} = \{m_1, m_2\}$ y $m_{obra2} = \{m_3, m_4\}$. El procesamiento de las obras del proyecto se pueden realizar al mismo tiempo; y no existe ninguna precedencia entre las obras del proyecto, por lo tanto se puede considerar cualquier orden de procesamiento de las obras.

A. Revisión Bibliográfica

En la literatura se han propuesto nuevas versiones del algoritmo PSO, como por ejemplo en el estudio de Yang y Simon [5], donde proponen un nuevo método de actualización de la posición de las partículas considerando la peor posición individual y la peor posición grupal. Esto con el fin de explorar el espacio de soluciones, evitando que el algoritmo quede atrapado en un óptimo local en un pequeño número de iteraciones.

Richards y Ventura [6] proponen una configuración de búsqueda para mejorar la exploración del espacio, la cual consiste en la partición el espacio de soluciones en subespacios donde su centro de gravedad representa la mejor posición individual de una partícula. La partición del espacio se determina según el tamaño del enjambre de partículas. El problema de esta técnica radica en que entre más partículas tenga el enjambre, se requiere mayor tiempo computacional para realizar la partición del espacio de búsqueda.

El trabajo de Dong y Qiu [7], considera un método híbrido entre *Particle Swarm Optimization* (PSO) y la metaheurística de enfriamiento simulado a través de un factor de inercia w que se involucra en la ecuación de cambio de posición de la partícula. Mediante la probabilidad de aceptación basada en la distribución de Boltzman-Gibbs, se determina si en una iteración del PSO se debe considerar el

uso del factor de inercia para el control de la tasa de cambio de las partículas. Chaojun y Zulian [8], emplearon un método híbrido similar al de Dong y Qiu [7], para analizar la eficiencia del método con respecto a la calidad de la solución en problemas de optimización sin restricciones. Jiao et al. [9] consideraron que el factor de inercia que controla la tasa de cambio de las partículas debe variar según la iteración del algoritmo y la mejor posición grupal.

Esto permite que en las primeras iteraciones del algoritmo se realice una exploración global del espacio de búsqueda y después se proceda con una exploración local antes de que se cumpla el criterio de parada del algoritmo. Por otro lado, Bergh y Engelbrecht [10], realizaron un estudio de las trayectorias que deben recorrer las partículas del enjambre según las condiciones del problema a resolver con esta metaheurística. Se consideró el factor inercial aplicado a la tasa de cambio de la posición y el factor de constricción aplicado a la actualización de la posición de las partículas. Se concluyó que según la forma como se explore el espacio de búsqueda, el algoritmo siempre converge a un punto estable (óptimo local), dependiendo del tamaño del enjambre y las dimensiones del espacio de búsqueda.

Con respecto a la metaheurística *Discrete Particle Swarm Optimization* (DPSO); esta se ha aplicado en problemas de programación lineal entera [11] y en problemas de secuenciación de tareas en una máquina dependiendo del tiempo de alistamiento para minimizar la tardanza ponderada total [12].

Villoria y Pastor [13], consideran una estrategia de diversificación de búsqueda en el algoritmo DPSO, mediante la definición de un componente aleatorio en el cambio de posición de las partículas; el valor que adquiere el parámetro cognitivo aleatorio es vital en la prevención de una convergencia prematura de la solución en el algoritmo.

Con respecto a los sistemas de producción con máquinas en paralelo, éstos se pueden clasificar en máquinas idénticas, uniformes y no idénticas, de acuerdo a la relación entre el tiempo de procesamiento de los trabajos y las máquinas del sistema [14].

Sistema de producción con máquinas en paralelo

Con respecto a los sistemas de producción con máquinas idénticas, se han desarrollado métodos exactos para los problemas que requieren la minimización de la tardanza total de trabajos. Biskup et al. [15] y Sidhoum et al. [16] proponen modelos de programación lineal que junto con métodos heurísticos, permiten resolver problemas de gran escala.

También se han propuesto técnicas metaheurísticas para resolver problemas de programación con máquinas idénticas en paralelo. Bilge et al. [17], propusieron una Búsqueda Tabú (BT) para el problema de minimizar la tardanza total, el cual se basa en heurísticas de lista. Las heurísticas de lista se basan en la ordenación de los trabajos mediante reglas de despacho; las cuales, deben tener relación directa con la reducción de la tardanza total.

Anghinolfi y Paolucci [18] consideraron una metaheurística híbrida para minimizar la tardanza en máquinas idénticas en paralelo. Esta metaheurística consiste en la unión de los criterios de movimiento tabú, de la Búsqueda Tabú (BT); selección aleatoria de una solución basado en un valor de probabilidad del Enfriamiento Simulado (ES) y la búsqueda aleatoria en la vecindad. Estos criterios permiten una mejor exploración del espacio de soluciones.

Kashan y Karimi [19] proponen una adaptación del algoritmo de optimización de enjambre de partículas discreto (DPSO), mediante la representación de la solución como un vector que contiene información de las máquinas que se utilizan para procesar los trabajos, con el objetivo de minimizar el tiempo de terminación de todos los trabajos en un sistema de producción con máquinas idénticas en paralelo.

Sistema de producción con máquinas uniformes

En ambientes de producción con máquinas uniformes, Joo [14], considera varios métodos heurísticos basados en una programación lineal para los casos en que se presenten restricciones de tiempo de inicio de los trabajos, tiempos de

procesamientos idénticos y ventanas de tiempo para la disponibilidad de las máquinas. Por otro lado, Balakrishnan [20], proponen un modelo de programación lineal para el problema de máquinas uniformes con secuencias dependientes del tiempo de alistamiento; para problemas de gran escala, procede con la definición de procedimientos heurísticos basados en la programación lineal propuesta.

Sistema de producción con máquinas multiprocesadoras

Los sistemas multiprocesadores es un concepto relativamente nuevo dentro de la teoría de la programación de producción, el cuál está basado en sistemas de computadores en paralelo. Drozdowski [4], presenta una revisión del estado del arte para sistemas donde los trabajos requieren del uso de varias máquinas en el mismo instante de tiempo, realizando la diferencia entre máquinas idénticas y máquinas especializadas para una actividad; adicionalmente, propone una nomenclatura para este tipo de problemas en la programación de producción. Por otro lado, Zinder et.al [21], presenta un estudio sobre la complejidad de los sistemas multiprocesadores considerando restricciones de precedencias, elegibilidad de máquinas, tiempos de inicio de los trabajos y el número de máquinas disponibles para la realización de un trabajo en particular.

Con respecto a la obtención de soluciones en problemas de producción basados en sistemas de multiprocesadores, se han desarrollado métodos de solución mediante programación lineal permitiendo la interrupción o paradas en el proceso y los tiempos de alistamiento [22].

Otros autores han desarrollado métodos heurísticos para resolver problemas de sistemas multiprocesadores con restricciones de interrupción de actividades. Bianco et.al [23], proponen heurísticas basadas en restricciones lineales, para la reducción del retardo de los trabajos en sistemas multiprocesadores que permiten la interrupción durante el proceso de producción. Amoura et.al [24], implementan heurísticas junto con un análisis de complejidad en problemas de programación con

multiprocesadores independientes que deben procesar trabajos con o sin interrupción, para minimizar el tiempo de procesamiento de todos trabajos.

B. Discrete Particle Swarm Optimization

El algoritmo de Optimización de Enjambre de Partículas Discreto “Discrete Particle Swarm Optimization” (DPSO) desarrollado por Kennedy y Eberhart [25], el cuál se deriva del algoritmo de Optimización de Enjambre de Partículas “Particle Swarm Optimization” (PSO) desarrollado por los mismo autores en 1995; es una metaheurística que se basa en el comportamiento social de un grupo de organismos, como por ejemplo, el comportamiento de una manada de aves o peces en búsqueda de alimento.

Según el estudio de Kennedy y Eberhart [26], debe existir una sincronización en el movimiento de los organismos (partículas) que conforman el enjambre. Cada partícula cambia su posición a través de de una velocidad (Tasa de cambio de posición), la cual se basa en dos características esenciales: un factor de conocimiento cognitivo de su mejor posición durante el recorrido, y un factor de conocimiento social de la mejor posición obtenida en el grupo.

Suponer que en un espacio de soluciones bidimensional (x, y) , se dispone de un enjambre de diez partículas, las cuales viajan entre posiciones discretas del espacio sin conocer la posición que corresponde a la solución óptima, ver Figura 3.

La actualización de la posición de cada partícula en el espacio discreto de soluciones, se realiza mediante un criterio estocástico de selección establecido en (1).

$$x_{ki} = \begin{cases} 1 & \delta < P(\Delta x_{ki}) \\ 0 & d.l.c \end{cases} \quad (1)$$

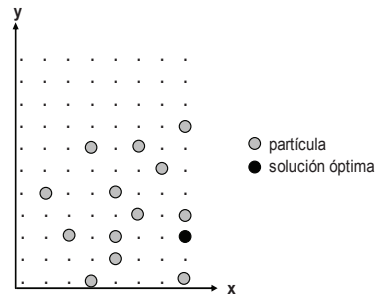


Figura 3. Posición bidimensional del enjambre de partículas en un espacio discreto de solución

donde $P(\Delta x_{ki})$ es la probabilidad de realizar el cambio de posición de la partícula k en la dimensión i , y δ es un valor aleatorio en el dominio $[0,1]$. El cambio de posición de la partícula depende de su propia experiencia o *factor cognitivo* y de la mejor experiencia del resto de partículas del enjambre o *factor de conocimiento social* [27]. Una posible actualización de las partículas en el espacio de soluciones se presenta en la Figura 4.

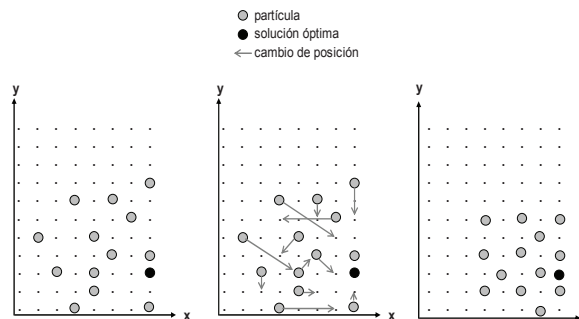


Figura 4. Posición bidimensional del enjambre de partículas en un espacio discreto de solución.

El proceso continúa hasta que se cumpla con el criterio del número de iteraciones del algoritmo, este criterio puede ser entendido como el tiempo en que las partículas del enjambre recorren el espacio de soluciones.

La metaheurística DPSO aplicada al problema de programación de máquinas uniformes en paralelo, puede ser entendida mediante la siguiente analogía con respecto a los conceptos básicos de optimización, tal como se presenta en la Tabla 4.

Generalmente, el algoritmo DPSO sigue la misma estructura de PSO; la diferencia radica en la forma de actualizar la posición de la partícula. Para el caso de DPSO que trabaja con soluciones discretas, Kennedy y Eberhart (1997) propusieron una transformación logística para determinar la probabilidad de aceptar o no el cambio de la posición de una partícula $\rho(\Delta x_{rd}^{t+1})$. Para ellos, la posición de la partícula sólo puede adoptar valores binarios.

TABLA IV
ANALOGÍA ENTRE DPSO Y OPTIMIZACIÓN

Mejor posición de la partícula en el espacio de búsqueda	Optimización
Partícula	Solución
Enjambre	Conjunto de soluciones
Posición de la partícula	Función Objetivo
Cambio de posición de la partícula (Velocidad)	Generación de una solución
Nueva posición de las partículas que constituyen el enjambre	Nuevo conjunto de soluciones
Factor cognitivo de la partícula	Estrategia de diversificación en la búsqueda de la solución
Factor social de la partícula	Estrategia de intensificación en la búsqueda de la solución

La actualización de la posición de las partículas para el caso discreto, se representa por (2) y (3):

$$\rho(\Delta x_{ld}^{t+1}) = \frac{1}{1 + e^{-\left(\Delta x_{ld}^{t+1}\right)}} \quad (2)$$

$$x_{ld}^{t+1} = \begin{cases} x_{ld}^{t+1} = 1 & \text{si } rand() < P(\Delta x_{ld}^{t+1}) \\ x_{ld}^{t+1} = 0 & \text{d.l.c} \end{cases} \quad (3)$$

donde $rand()$ es un valor aleatorio en el intervalo de $[0, 1]$. Al igual que el algoritmo PSO, el criterio de parada en la mayoría de los casos es el de llegar al número máximo de iteraciones permitido en el algoritmo. Para el caso de la programación de producción, considerar que la partícula l , representa la solución del problema y las dimensiones que definen el espacio de

soluciones donde viaja la partícula, representan los elementos principales de la programación; que en este caso son las máquinas, los proyectos y las obras de construcción.

IV. PROPUESTA DE SOLUCIÓN

Inicialmente, se considera una heurística basada en la velocidad de procesamiento de cada máquina y el tiempo de inicio de cada proyecto, para obtener una solución inicial factible. Posteriormente, se procede con esta solución inicial para la generación del enjambre de partículas (conjunto de soluciones factibles), las cuales se generan mediante un criterio estocástico que permite el cambio aleatorio en la asignación de máquinas a los proyectos de construcción.

Seguidamente, se comienza con las iteraciones del algoritmo DPSO hasta que se cumpla el criterio del máximo número de iteraciones. La mejor solución obtenida es considerada para la representación gráfica de la programación de máquinas mediante un diagrama Gantt. Finalmente, se considera técnicas de reprogramación basado en la llegada de nuevos proyectos de construcción, respetando la programación de obras que se están procesando hasta el instante de tiempo en que se debe comenzar un nuevo proyecto.

A. Solución del Problema

En el algoritmo propuesto se ha considerado el uso de la representación binaria para la asignación de máquinas uniformes a las obras que constituyen un proyecto en particular. Se define la partícula l en la iteración t como X_l^t ; cada partícula se encuentra en un espacio tridimensional de soluciones binarias. Dicho espacio está constituido por la dimensión máquina i , proyecto j , y obra k ; tal como se presenta en (4):

$$(4) \quad x_{lijk}^t = \begin{cases} 1 & \text{si en la iteracion } t, \text{ el elemento de la particula } l \\ & \text{indica que la maquina } i \text{ es asignada para} \\ & \text{procesar la obra } k \text{ del proyecto } j \\ 0 & \text{d.l.c} \end{cases}$$

Sea ϕ_{ik} una matriz binaria que identifica si la máquina i se especializa en procesar la obra k , y π_{jk} una matriz binaria que informa si la obra k es requerida en el proyecto j . Al considerar estas dos matrices de información, y la cantidad máxima de maquinas permitidas para procesar la obra k del proyecto j , $side_{kj}$; en este documento, se propone la siguiente heurística de asignación de máquinas, basado en un criterio de selección estocástico:

Paso 1: Sea M , el conjunto de máquinas disponibles; y N el conjunto de proyectos que se debe procesar. Ordenar en forma ascendente los tiempos de inicio de los proyectos; seguidamente, identificar para una obra k en particular, si se cumple $\phi_{ik} = 1 \quad \forall i \in M$ y si $\pi_{jk} = 1 \quad \forall j \in N$.

Paso 2: Al cumplirse el Paso 1, se procede con el cálculo de λ , que representa la probabilidad de éxito de que una máquina i procese el proyecto j (5).

$$(5) \quad \lambda = \frac{1}{1 + e^{-\left(\frac{i}{M}\right)\left(\frac{j}{N}\right)}}$$

Se considera el uso de la regresión logística para calcular el valor de probabilidad λ , puesto que el campo de variación de la relación logarítmica entre la probabilidad de éxito y de fracaso es el conjunto de todos los número reales. [25]. El criterio estocástico se presenta en (6).

$$(6) \quad x_{lijk}^t = \begin{cases} 1 & \text{si } \lambda \leq \theta \\ 0 & \text{d.l.c} \end{cases}$$

donde θ es un valor aleatorio en el intervalo $[0,1]$.

Paso 3: Sea O_{jk}^t el número de máquinas disponibles para procesar la obra k del proyecto j durante la iteración t . La asignación de máquinas para una obra en particular se obtiene mediante la ecuación (7):

$$(7) \quad O_{jk}^t = \sum_{i=1}^m x_{lijk}^t \quad \begin{matrix} \forall j \in N \\ \forall k \in K \\ \forall l \in L \end{matrix}$$

Si la asignación cumple con la expresión (8), se continúa con el Paso 6.

$$(8) \quad 0 < O_{jk}^t \leq side_{kj} \quad \begin{matrix} \forall j \in N \\ \forall k \in K \end{matrix}$$

Paso 4: Si en el paso anterior se obtiene que $O_{jk}^t = 0$, entonces se debe retornar al Paso 2, hasta que se cumpla (8); en caso contrario, si $O_{jk}^t > side_{kj}$, se procede con el Paso 5.

Paso 5: Como $O_{jk}^t > side_{kj}$, se aplica el criterio estocástico presentado en (9), con el propósito de actualizar del número de máquinas disponibles para procesar la obra k del proyecto j , teniendo en cuenta que $\phi_{ik} = 1 \quad \forall i \in M$.

$$(9) \quad O_{jk}^t = \begin{cases} O_{jk}^t - 1 \wedge x_{lijk}^t = 0 & \text{si } \theta_1 \leq \lambda \\ O_{jk}^t & \text{d.l.c} \end{cases}$$

donde θ_1 es un valor aleatorio en el intervalo $[0,1]$. Este proceso continua hasta que se cumple con la expresión (8). Seguidamente, se procede con el Paso 6.

Paso 6: Si $x_{lijk}^t = 1$, se procesa la obra k del proyecto j en la máquina i . Sea S_{li}^t , el tiempo de ocupación de la máquina i en la partícula l de la iteración t ; y c_{ij}^t , el tiempo de procesamiento del proyecto j en la máquina i en la partícula l de la

iteración t . Para calcular estos tiempos, se debe considerar los siguientes casos:

Caso I. Si $x_{lijk}^t = 1$ y $S_{li}^t = 0$, el tiempo de ocupación de la máquina i , se calcula mediante la ecuación (10):

$$S_{li}^t = r_j + \left[\frac{P_j}{O_{jk}^t v_i} \right] \quad \forall i \in M \quad (10)$$

y el tiempo de procesamiento del proyecto j en la máquina i , se obtiene mediante la ecuación (11):

$$c_{lij}^t = S_{li}^t \quad (11)$$

La representación gráfica del tiempo de ocupación de la máquina i se presenta en la Figura 5.

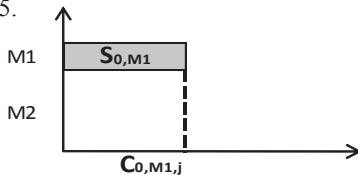


Figura 5. Tiempo de ocupación de la máquina $M1$ para procesar la obra k del proyecto j , cuando se cumple el caso I.

Caso II. Si $x_{lijk}^t = 1$; $S_{li}^t \neq 0 \wedge S_{li}^t \leq r_j$, el tiempo de ocupación de la máquina i para procesar el proyecto j , se calcula mediante la ecuación (12):

$$S_{li}^t = S_{li}^t + \left[r_j - S_{li}^t \right] + \left[\frac{P_j}{O_{jk}^t v_i} \right] \quad \forall i \in M \quad (12)$$

y el tiempo de procesamiento del proyecto j en la máquina i , se obtiene mediante la ecuación (11). La representación gráfica del tiempo de ocupación de la máquina i se presenta en la Figura 6.

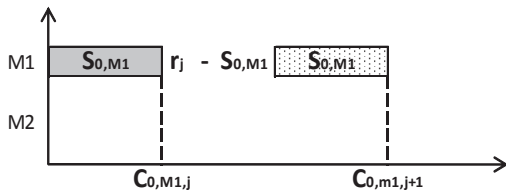


Figura 6. Tiempo de ocupación de la máquina $M1$ para procesar la obra k del proyecto $j+1$, cuando se cumple el caso II

Caso III. Si $x_{lijk}^t = 1$; $S_{li}^t \neq 0 \wedge S_{li}^t > r_j$, el tiempo de ocupación de la máquina i para procesar el proyecto j , se calcula mediante la ecuación (13):

$$S_{li}^t = S_{li}^t + \left[\frac{P_j}{O_{jk}^t v_i} \right] \quad \forall i \in M \quad (13)$$

y el tiempo de procesamiento del proyecto j en la máquina i , se obtiene mediante la ecuación (11). La representación gráfica del tiempo de ocupación de la máquina i se presenta en la Figura 7. Este procedimiento se repite hasta que todas las obras de todos los proyectos se programen en las máquinas disponibles.

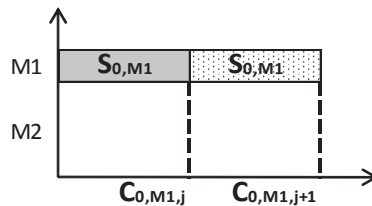


Figura 7. Tiempo de ocupación de la máquina $M1$ para procesar la obra k del proyecto $j+1$, cuando se cumple el caso III

Ahora, con el propósito de calcular la tardanza ponderada total de la programación de proyectos de la partícula l en la iteración t , se considera la ecuación (14).

$$T_l^t = \sum_{j=1}^N w_j \left\{ \max_{i \in M} (c_{lij}^t) - d_j \right\} \quad \forall l \in L \quad (14)$$

El siguiente ejemplo está basado en información parcial suministrada por la empresa, los nombres de los proyectos, así como los nombres de los ingenieros fueron cambiados y se omite información respecto a las utilidades esperadas de los proyectos por razones de confidencialidad de la empresa. Suponer que la empresa dispone de cuatro máquinas para el procesamiento de pilotes y pantallas.

Maquina	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
TH-15						San Alonso					Sevilla 2					Héroes									
						Pilotes k					Pilotes k					Pilotes k									
						Carlos Mora					Edwing					Wolfgang									
RS-18						Edificio ABC Ltda					Sevilla 2					Edificio San Agustín									
						Pilotes k					Pilotes k					Pilotes k									
						Freddy P					Edwing					Luisa M									
LIEBHERR						San Alonso					Edificio San Agustín														
						Pantallas					Pantallas														
						Carlos Mora					Luisa M														
C8						Edificio ABC Ltda					San Alonso														
						Pantallas					Pantallas														
						Freddy P					Carlos Mora														

Figura 8. Representación gráfica de la solución inicial del problema

TABLA V
INFORMACIÓN DE LAS MÁQUINA PARA EL PROCESAMIENTO DE OBRAS DE CONSTRUCCIÓN

Máquina	Capacidad Máxima	Pilotes	Pantallas
TH-15	2	X	
RS-18	2	X	
LIEBHERR	2		X
C8	2		X

La Tabla 5, presenta la información de la especialidad de cada máquina y la capacidad máxima diaria de procesamiento.

La empresa tiene cinco proyectos de construcción; los cuales tienen la misma importancia en ser procesados $w_j=1$. La información de las obras que se requieren en los proyectos, el ingeniero encargado, el tiempo de inicio, terminación y de procesamiento (en días) de éstas; se presenta en la Tabla 6.

Considerar que tanto las máquinas especializadas para el procesamiento de pilotes como las máquinas especializadas para el procesamiento de pantallas, se pueden asignar en la obra de un proyecto en particular, $Side_{jk} = 2$. Para obtener la solución inicial del problema, se consideran los pasos descritos anteriormente. La representación de la solución inicial, teniendo en cuenta que el orden de los sub-índices en función del menor tiempo de inicio es $j = \{ABC, San$

$Alonso, Sevilla 2, San Agustín, Héroes\}$ y $k = \{Pilotes, Pantallas\}$; se presenta en (15).

$$x_{01jk}^0 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, x_{02jk}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}, x_{03jk}^0 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, x_{04jk}^0 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (15)$$

Al sumar las cuatro matrices que representan la solución inicial, se obtiene la matriz O_{jk}^0 , la cuál representa el número de máquinas disponibles para procesar la obra k del proyecto j , tal como se presenta en (16).

$$O_{jk}^0 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \quad (16)$$

Se puede observar que los elementos de esta matriz cumplen con la restricción establecida en (8). La representación gráfica de la solución inicial se muestra en la Figura 8.

El tiempo total de procesamiento de los proyectos son $c_{ABC} = 10$ días, $c_{San Alonso} = 15$ días, $c_{Sevilla} = 19$ días, $c_{San Agustín} = 22$ días y $c_{Héroes} = 25$ días. Al calcular la tardanza ponderada total mediante (14), se obtiene que ningún proyecto se encuentre tardío; luego se puede cumplir con las fechas de terminación establecidas entre la empresa y los clientes

TABLA VI. INFORMACIÓN DE LOS PROYECTOS DE CONSTRUCCIÓN

Proyecto	Ingeniero	Inicio	Terminación	Pilotes	Procesamiento	Pantallas	Procesamiento
ABC Ltda.	Freddy P	3	25	X	14	X	14
San Alonso	Carlos Mora	5	30	X	20	X	20
Sevilla 2	Edwing G	7	38	X	15	----	----
San Agustín	Luisa M	9	30	X	15	X	15
Héroes	Wolfgang	10	30	X	12	----	----

La empresa requiere una reprogramación de los proyectos debido al compromiso de realizar nuevos proyectos de construcción. A continuación, se presenta el procedimiento para la reprogramación de proyectos:

Paso 1: Sea Q el conjunto que contiene los proyectos programados, y Q' el conjunto que contiene los nuevos proyectos a programar. De los nuevos proyectos que no han sido programados, se identifica el menor tiempo de inicio r_j ; con el propósito de definir el tiempo en que se debe realizar la reprogramación de los proyectos de construcción, ψ (17).

$$\psi = \min_{j \in N} \{r_j / j \in Q'\} \quad (17)$$

Paso 2: Al obtener la información del tiempo en que se debe proceder con la reprogramación, y considerar al conjunto de los proyectos programados Q ; se definen los siguientes casos para conocer las obras de los proyectos que deben respetarse en la reprogramación:

Caso I. Si $j \in Q$, $x_{ijk}^t = 1$, $S_{ii}^t \neq 0 \wedge S_{ii}^t < r_j$; se debe evaluar si la obra k del proyecto $j+1$ se programa o no en la máquina i . El tiempo de ocupación en la máquina i se calcula mediante (12). La Figura 6, presenta la condición establecida en este caso.

La máquina $M1$, se especializa en procesar la obra k . Ahora, mediante el tiempo de inicio de la reprogramación encontrado en el Paso I, se procede con la decisión de aceptar o no la obra k del proyecto $j+1$, en la reprogramación.

Si la obra k del proyecto $j+1$, cumple que $\psi \leq r_j - S_{0,M1}$, se debe considerar dicha obra en el proceso de reprogramación; en caso contrario, la obra será respetada durante el proceso de reprogramación. La Figura 9 presenta la decisión de aceptar o no a la obra k en el proceso de reprogramación.

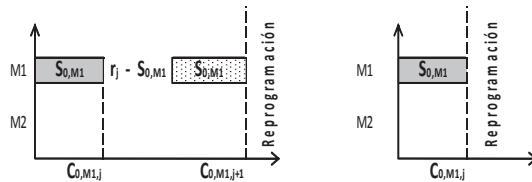


Figura 9. Decisión de reprogramar la obra k del proyecto $j+1$, a) se respeta la asignación inicial de la obra k , cuando el tiempo de inicio de la reprogramación es mayor que el tiempo de inicio de la obra k , y b) se considera la obra k en el proceso de reprogramación, cuando el tiempo de inicio de la reprogramación es menor o igual al tiempo de inicio de la obra k .

Caso II. Si $j \in Q$, $x_{ijk}^t = 1$; $S_{ii}^t \neq 0 \wedge S_{ii}^t > r_j$; se debe evaluar si la obra k del proyecto $j+1$ se programa o no en la máquina i . El tiempo de ocupación de la máquina i , se calcula mediante la expresión (13). La [Figura 7], presenta la condición establecida en este caso.

La máquina $M1$, se especializa en procesar la obra k . Al considerar el tiempo de inicio de la reprogramación ψ , se analiza si se debe involucrar la obra k del proyecto $j+1$ en la reprogramación.

Si la obra k del proyecto j cumple que $\psi \geq C_{0,M1,j+1}$, se debe considerar que la obra k no se tendrá en cuenta en la reprogramación; en caso contrario, la obra debe considerarse en la reprogramación. La Figura 10 presenta el criterio de aceptar o no a la obra k en la reprogramación.

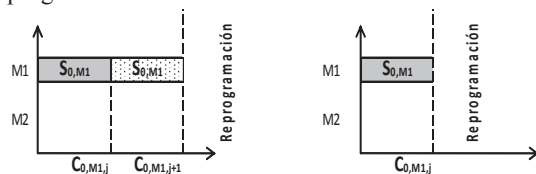


Figura 10. Decisión de reprogramar la obra k del proyecto $j+1$, a) se respeta la asignación inicial de la obra k , cuando el tiempo de inicio de la reprogramación es mayor que el tiempo de procesamiento de la obra k , y b) se considera la obra k en el proceso de reprogramación, cuando el tiempo de inicio de la reprogramación es menor o igual al tiempo de procesamiento de la obra k .

Paso 3: Las obras de los proyectos que pertenecen al conjunto Q , y que fueron respetadas en el proceso de reprogramación, indica que el elemento de la partícula t durante la iteración j , el cuál informa si la obra k del proyecto j es asignada a la máquina i , conserva el valor asignado en la programación original; mientras que las demás obras que se tienen en cuenta en la reprogramación cambian su valor, tal como indica la expresión (18).

$$x'_{ijk} \begin{cases} 1 & \text{si } \psi > r_j - S_{0,M1}^0 \vee \psi \geq C_{0,M1,j+1}^0 \\ 0 & \text{si } \psi \leq r_j - S_{0,M1}^0 \vee \psi < C_{0,M1,j+1}^0 \end{cases} \quad (18)$$

En el algoritmo de reprogramación propuesto en este documento, se consideró que el número de máquinas disponibles para procesar la obra k del proyecto j , O'_{jk} se conserva durante el proceso; puesto que el traslado de varias máquinas de un proyecto j a un proyecto $j+1$, produce pérdida de tiempo en el cumplimiento de la fecha de terminación de los nuevos proyectos.

Paso 4: La heurística de solución inicial y su posterior mejoramiento a través de la metaheurística DPSO, se aplica a las obras de los proyectos que pertenecen al conjunto Q' y las obras de los proyectos que pertenecen al conjunto Q , y que según el Paso 2, se tienen en cuenta en la reprogramación. Una vez terminadas las iteraciones de la metaheurística, se obtiene la solución para el problema de reprogramación de multiprocesadores uniformes en proyectos de construcción.

V. RESULTADOS COMPUTACIONALES

El algoritmo DPSO propuesto fue codificado en lenguaje de programación Visual Basic for Applications de Microsoft Excel y las corridas fueron realizadas en un computador Intel Pentium IV de 3 GHz y 992 MB de memoria RAM. El enjambre de partículas discreto fue implementado para los

problemas $Q//Cmax$ y $Q//\sum T$ en Visual Basic, con el propósito de evaluar la calidad de la solución del algoritmo propuesto con respecto al valor de la función objetivo.

A. Parámetros del Algoritmo

El parámetro cognitivo c_1 y social c_2 , los cuales son requeridos en el algoritmo de enjambre de partículas discreto, fueron calibrados de acuerdo a un diseño experimental por bloques; por lo tanto para el problema $Q//Cmax$, $c_1 = 2.5$ y $c_2 = 2$ es la combinación de parámetros adecuada en la obtención de soluciones. Con respecto al problema $Q//\sum T$, la menor varianza del error obtenida en la experimentación, se obtiene que $c_1 = 2$ y $c_2 = 2$, es la combinación del parámetros adecuada para dicho problema.

B. Resultados computacionales para el problema $Q//Cmax$

Para el problema $Q//Cmax$, se considera que las máquinas tienen una velocidad de procesamiento equivalente a la unidad, $v = 1$. Se construyeron veintisiete instancias para analizar la calidad de la solución de la metaheurística; los tiempos de procesamiento se estimaron mediante la distribución uniforme. La Tabla 7 presenta el resumen de los criterios considerados para la generación de las veintisiete instancias.

TABLA VII
RESUMEN DE LA GENERACIÓN DE INSTANCIAS PARA EL PROBLEMA $Q//Cmax$

Máquinas	Trabajos	Procesamiento
3, 5, 8	10,25,50	$U(5,15)$
		$U(10,30)$
		$U(10,50)$

Con el propósito de obtener la solución óptima de las veintisiete instancias generadas, se consideró el modelo de programación lineal entero binario, presentado en el trabajo de Kashan y Karimi [19]; el modelo fue implementado en el programa de optimización Xpress. Con respecto a la solución obtenida mediante DPSO, se considera que el trabajo j no está compuesto de operaciones; es decir, que el proyecto j no está constituido por obras, y se requiere de una sola máquina para su procesamiento, $O_j = 1$. Los resultados se presentan en el Anexo I.

Como se puede observar en el Anexo I, el comportamiento del algoritmo DPSO en pequeñas y medianas instancias es aceptable, ya que en instancias pequeñas se logra un gap promedio del 1%. Para las medianas instancias que disponen de cinco y ocho máquinas, el comportamiento promedio con respecto a la solución óptima es del 5,26 %. El incremento del promedio del gap, implica que ante el incremento de las dimensiones del espacio de búsqueda, el enjambre de partículas discreto propuesto en este documento, requiere de más partículas y mayor tiempo computacional de exploración del espacio para encontrar soluciones muy cercanas al óptimo. El tiempo computacional promedio requerido para obtener las soluciones de las instancias del problema $Q//Cmax$ con el algoritmo DPSO es de 3 minutos.

C. Resultados computacionales para el problema $Q//\sum T$

Para el problema $Q//\sum T$, se considera que las máquinas tienen una velocidad de procesamiento equivalente a la unidad, $v = 1$ y que todos los trabajos tienen igual importancia $w_j = 1$. Se construyeron dieciocho instancias para analizar la calidad de la solución de la metaheurística; los tiempos de procesamiento se estimaron mediante la distribución uniforme. La Tabla 8 presenta el resumen de los criterios considerados para la generación de las dieciocho instancias.

TABLA VIII
RESUMEN DE LA GENERACIÓN DE INSTANCIAS PARA EL PROBLEMA $Q//\sum T$

Máquinas	Trabajos	Procesamiento
3, 5, 8	10,25	$U(5,15)$
		$U(10,30)$
		$U(10,50)$

Siguiendo la propuesta de Hariri y Potts [28], la fecha de terminación de los trabajos para cada instancia, fue generada mediante la expresión (19):

$$d_j = p_j + U(0, \lfloor 0.2P \rfloor) \tag{19}$$

donde d_j representa la fecha de terminación del trabajo j ; p_j , es el tiempo de procesamiento del trabajo j , y P , es el tiempo de procesamiento promedio de los trabajos.

Para obtener la solución óptima de las dieciocho instancias generadas, se consideró el modelo de programación lineal entero mixto, propuesto en el trabajo de Biskup et al. [15]; el modelo fue implementado en el programa de optimización Xpress. Con respecto a la solución obtenida mediante DPSO, se considera que el trabajo j no está compuesto de operaciones; es decir, que el proyecto j no está constituido por obras, y se requiere de una sola máquina para su procesamiento, $O_j = 1$. Los resultados se presentan en el Anexo II.

Como se puede observar en los resultados de el Anexo II para el problema $Q//\sum T$ se presenta el mismo comportamiento descrito en la sección anterior. En instancias pequeñas, el algoritmo DPSO obtiene soluciones muy cercanas al óptimo con un gap promedio de 2,4 %; mientras que para medianas instancias que requieren cinco y ocho máquinas, el gap es de 6,08 %. El tiempo computacional promedio requerido en la obtención de soluciones para la instancias del problema $Q//\sum T$, es de 4 minutos.

La representación binaria y el criterio estocástico propuesto en este trabajo, según los resultados obtenidos para las instancias de experimentación, permite encontrar buenas soluciones en problemas que consideren la asignación de máquinas en obras de construcción considerando elegibilidad de máquinas y tiempo de inicio de los proyectos.

VI. CONCLUSIONES

El principal aporte de este trabajo es el de resolver un problema de la vida real considerando los fundamentos del estado del arte de la programación de producción. Se ha implementado un esquema de representación binaria, basado en un criterio estocástico, para resolver el problema de programación de máquinas multiprocesadoras uniformes en proyectos de construcción con el objetivo de minimizar tanto el tiempo de terminación de todos los proyectos, como la tardanza de dichos proyectos con base en niveles de utilidad esperada. Adicionalmente, proveer un programa que permita la utilización de esta técnica de solución en la empresa considerada como caso de estudio, teniendo en cuenta el esquema de la reprogramación de máquinas, ya que la asignación de máquinas para el procesamiento de las obras de construcción se realiza en forma manual y requiere de varios días para obtener una programación estándar de las máquinas.

Con base en este trabajo, posibles investigaciones futuras podría ser la aplicación de este algoritmo en otros ambientes de producción, considerar tiempos de alistamiento de las máquinas para el procesamiento de obras de construcción, implementar otros criterios estocásticos que permitan la generación de soluciones factibles para el problema de máquinas multiprocesadoras uniformes, diseño de otras versiones basadas en la metaheurística de enjambre de partículas discreto en otros problemas relacionados con la programación de producción. Adicionalmente, implementar este algoritmo en otros problemas de la industria como en el sector de petróleos donde se requiere la asignación de máquinas especializadas en proyectos de yacimientos, cimentación, entre otros; en el sector metalmecánico donde se disponen de máquinas especializadas para la fabricación de herramientas; y en el procesos de tratamientos térmicos, donde se debe asignar de manera eficiente los dispositivos y hornos necesarios para el tratamiento de piezas de acero.



Orlando Federico González Casallas, Lugar de nacimiento Bogotá, D.C; Magister en Ingeniería Industrial, con énfasis en Sistemas de Producción y Logística, Universidad de los Andes (2009); Ingeniero Industrial, Universidad Libre de Colombia (Seccional Cúcuta) (2006). Docente de Tiempo Completo de la facultad de Ingeniería Industrial, Universidad Pontificia Bolivariana (Seccional Bucaramanga). Interés en investigaciones relacionadas con Programación de Producción, Modelos de Sistemas Logísticos, Pronósticos y Series de Tiempo, Estadística Paramétrica, Estadística Bayesiana y modelos matemáticos basados en Optimización Matemática e Investigación de Operaciones.

REFERENCIAS

- [1] Ramamritham, K. and Stankovic, J.A., Scheduling algorithms and operating systems support for real-time systems. *Proceedings of the IEEE*, 1994, 82, 55-67.
- [2] Graham, R.L., Lawler, E.L., Lenstra, J.K. and RinnooyKan, A.H.G., Optimization and approximation in deterministic sequencing and scheduling theory: A survey. *Annals of Discrete Mathematics*, 1979, 5, 287-326.
- [3] Veltman, B., Lageweg, B.J. and Lenstra J.K., Multiprocessor scheduling with communications delays. *Parallel Computing*, 1990, 16, 173-182.
- [4] Drozdowski, M., Scheduling multiprocessor tasks- an overview. *European Journal of Operational Research*, 1996, 94, 215-230.
- [5] Yang, C. and Simon D., A new particle swarm optimization technique. *18th International Conference on Systems Engineering*, 2005, 1, 164-169.
- [6] Richards, M., and Ventura, D., Choosing a Starting Configuration for Particle Swarm Optimization. *International Joint Conference on Neural Networks*, 2004, 67, 25-29.
- [7] Dong, C. and Qiu, Z., Particle Swarm Optimization Algorithm Based on the Idea of Simulated Annealing. *Automation Control Research Institute, China*, 2006.
- [8] Chaojun, D. and Zulian Q., Particle swarm optimization algorithm based on the idea of simulated annealing. *International Journal of Computer Science and Network Security*, 2006, 6, 152-157.
- [9] Jiao, B., Lian Z. and Gu X., A dynamic inertia weight particle swarm optimization algorithm. *Chaos, Solitons and Fractals*, 2008, 37, 698-705
- [10] Bergh, F. and Engelbrecht A.P., A study of particle swarm optimization particle trajectories. *Information Sciences*, 2006, 176, 937-971.
- [11] Laskari, E.C., Parsopoulos, K.E. and Vrahatis, M.N., Particle swarm optimization for integer programming. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Honolulu, Hawaii USA, 2002, 2, 1582-1587.
- [12] Tseng, C. and Liao, C., A particle swarm optimization algorithm for hybrid flow-shop scheduling with multiprocessor tasks. *International Journal of Production Research*, 2008, 46, 4655-4670.
- [13] Villoria A.G. and Pastor R., Introducing dynamic diversity into a discrete particle swarm optimization. *Computers & Operations Research* (2007), doi:10.1016/j.cor.2007.12.001
- [14] Joo, U.G., job assignment algorithms on uniform parallel machines. *Engineering Optimization*, 2001, 33, 351-371.
- [15] Biskup, D., Herrmann, J. and Gupta, J. N.D., Scheduling identical parallel machines to minimize total tardiness. *International Journal of Production Economics* (2008), doi:10.1016/j.ijpe.2008.04.011.
- [16] Sidhoum, S.K., Solis Y.R. and Sourd F., Lower bounds for the earliness-tardiness scheduling problem on parallel machines with distinct due dates. *European Journal of Operational Research*, 2008, 189, 1305-1316.
- [17] Bilge, Ü., Kiraç F., Kurtulan M. and Pekkün P., A tabu search algorithm for parallel machine total tardiness problem. *Computer & Operations Research*, 2004, 31, 397-414.
- [18] Anghinolfi, D. and Paolucci, M., Parallel machine total tardiness scheduling with a new hybrid metaheuristic approach. *Computers & Operations Research*, 2007, 34, 3471-3490.
- [19] Kashan, A.H. and Karimi, B., A discrete particle swarm optimization algorithm for scheduling parallel machines, *Computers & Industrial Engineering* (2008), doi: 10.1016/j.cie.2008.05.007.
- [20] Balakrishnan, N., Kanet, J.J. and Sridharan S.V., Early/tardy scheduling with sequence dependent setups on uniform parallel machines. *Computers & Operations Research*, 1999, 26, 127-141.
- [21] Zinder, Y., Do V.H. and Oguz, C., Computational complexity of some scheduling problems with multiprocessor tasks. *Discrete Optimization*, 2005, 2, 391-408.
- [22] Boctor, F.F., Renaud, J., Ruiz, A. and Tremblay, S., Optimal and heuristic solution methods for a multiprocessor machine scheduling problem. *Computers and Operations Research* (2009), doi: 10.1016/j.cor.2008.12.017.
- [23] Bianco, L., Blazewicz, J., Dell'Olmo, P. and Drozdowski, M., Linear algorithms for preemptive scheduling of multiprocessor tasks subject to minimal lateness. *Discrete Applied Mathematics*, 1997, 72, 25-46.

[24] Amoura, A.K., Bampis, E., Kenyon, C. and Manoussakis, Y., Scheduling Independent Multiprocessor Tasks. *Algorithmica*, 2002, 32, 247-261.

[25] Kennedy, J. and Eberhart, R., A discrete binary version of the particle swarm algorithm. *IEEE International Conference on Computational Cybernetics and Simulation*, 1997, 5, 4104-4108.

[26] Kennedy, J. and Eberhart, R., Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*. Piscataway, NJ, IEEE service center, 1995, 4, 1942-1948.

[27] Shi, Y. and Eberhart, R.C., A modified particle swarm optimizer. *Proceedings of the IEEE Congress on Evolutionary Computation*, 1998, 69-73.

[28] Hariri, A.M.A. and Potts, C.N., A branch and Bound algorithm to minimize the number of late jobs in a permutation flow shop. *European Journal of Operational Research*, 1989, 38, 228-238.

ANEXO I. RESULTADOS COMPUTACIONALES PARA EL PROBLEMA $Q//C_{max}$

m	Procesamiento	Óptimo	DPSO	Gap(%)
10	U(5,15)	36	36	0,0
	U(10,30)	77	77	0,0
	U(10,50)	98	98	0,0
	Promedio			0,0
3	U(5,15)	83	85	2,4
	U(10,30)	167	168	0,6
	U(10,50)	283	290	2,5
	Promedio			1,8
50	U(5,15)	172	175	1,7
	U(10,30)	305	308	1,0
	U(10,50)	461	467	1,3
	Promedio			1,3
10	U(5,15)	23	24	4,3
	U(10,30)	51	55	7,8
	U(10,50)	62	65	4,8
	Promedio			5,7
5	U(5,15)	50	52	4,0
	U(10,30)	101	107	5,9
	U(10,50)	170	176	3,5
	Promedio			4,5
50	U(5,15)	103	108	4,9
	U(10,30)	183	190	3,8
	U(10,50)	279	285	2,2
	Promedio			3,6
10	U(5,15)	15	16	6,7
	U(10,30)	33	35	6,1
	U(10,50)	45	47	4,4
	Promedio			5,7
8	U(5,15)	32	34	6,3
	U(10,30)	63	66	4,8
	U(10,50)	106	108	1,9
	Promedio			4,3
50	U(5,15)	69	74	7,2
	U(10,30)	122	132	8,2
	U(10,50)	190	205	7,9
	Promedio			7,8

ANEXO II. RESULTADOS COMPUTACIONALES PARA EL PROBLEMA $Q//\Sigma T$

m	Procesamiento	Óptimo	DPSO	Gap(%)
10	U(5,15)	85	85	0,0
	U(10,30)	201	201	0,0
	U(10,50)	220	221	0,5
	Promedio			0,2
3	U(5,15)	142	150	5,6
	U(10,30)	340	362	6,5
	U(10,50)	365	371	1,6
	Promedio			4,6
10	U(5,15)	29	30	3,4
	U(10,30)	78	80	2,6
	U(10,50)	76	79	3,9
	Promedio			3,3
5	U(5,15)	50	55	10,0
	U(10,30)	132	138	4,5
	U(10,50)	126	130	3,2
	Promedio			5,9
10	U(5,15)	9	9	0,0
	U(10,30)	17	19	11,8
	U(10,50)	16	17	6,3
	Promedio			6,0
8	U(5,15)	14	15	7,1
	U(10,30)	28	30	7,1
	U(10,50)	23	26	13,0
	Promedio			9,1