

Secuenciación de operaciones para configuraciones de planta tipo flexible Job Shop: Estado del arte

Scheduling for shop settings types to flexible Job Shop: State of the art

Alexander Alberto Correa Espinal, Ph.D, Elkin Rodríguez Velásquez, MSc, María Isabel Londoño Restrepo, Ing.
Escuela de Ingeniería de la Organización, Universidad Nacional de Colombia - Sede Medellín
alcorrea@unal.edu.co, erodrigu@unal.edu.co, milondo0@unal.edu.co

Recibido para revisión: 2 de Octubre de 2008, Aceptado: 28 de Noviembre de 2008, Versión final: 22 de Diciembre de 2008

Resumen—Un número considerable de investigaciones han sido enfocadas en solucionar el problema de las configuraciones de planta tipo Flexible Job Shop, problema genérico correspondiente al problema Job Shop; por años, numerosos métodos han sido propuestos para resolver diferentes instancias del problema, dependiendo de las restricciones impuestas y del objetivo mismo. Recientemente muchos de esos acercamientos se han concentrados en técnicas conocidas como algoritmos interactivos de búsqueda local o meta heurísticas. En este artículo estamos interesados en ilustrar el estado actual de desarrollo de los métodos meta heurísticos, principalmente algoritmos genéticos existentes en la literatura para el problema genérico del Job Shop minimizando el makespan. Adicional a esto incluimos las posibles líneas de investigación en trabajos futuros.

Palabras Clave—Flexible Job Shop, Algoritmos genéticos, Secuenciación de operaciones, Makespan.

Abstract—A considerable number of research have been focused on solving the problem of the configurations of shop type Flexible Job Shop, it's the generic problem for the Job Shop problem, for years, many methods have been proposed to solve different instances of the problem, depending on the constraints and the performance indicator. Recently many of these approaches have focused on techniques known as interactive local search algorithms or heuristics. In this paper, we are interested in illustrating the current state of development of the heuristic methods, mainly genetic algorithms in the literature for the generic problem of Job Shop minimizing the makespan. Additional to this we include the possible lines of research in future works.

Keywords—Flexible Job Shop, Genetic algorithms, Scheduling, Makespan.

I. INTRODUCCIÓN

Urwich y Brech [1984] introducen referencias sobre la planificación de la producción, indican que toda aquella persona que esté familiarizada con la industria debe apreciar la enorme ventaja que se obtiene de un análisis de las necesidades, si es posible, para todo el año. Indican que cuando uno conoce, a intervalos regulares de tiempo, la cantidad de producción que es demandada, por lo tanto, a procesar, es posible desarrollar programas para la manufactura, que permitan a la compañía alcanzar un pleno rendimiento; que la producción se abarate enormemente; que no sea necesario mantener grandes stocks en tiempos excesivos; que el capital y el interés se economicen, en general que aumente la eficacia [1].

La planificación es un proceso complejo, en el que hay que tener en cuenta muchos factores y considerar ciertos objetivos. Precisamente por este motivo no es posible tenerlos en cuenta a la vez y, por lo tanto, se debe establecer una metodología racional si se desean obtener resultados útiles. Por eso, se consideran diferentes niveles de planificación relacionados entre sí, con distintos niveles de detalle, y plazos de tiempo para llevarlos a cabo.

La secuenciación de operaciones hace parte del nivel de planificación a corto plazo y principalmente, es una área de investigación desde hace más de cinco décadas, es una rama de la optimización combinatoria que ha desarrollado su propia metodología utilizando herramientas matemáticas y computacionales variadas; donde el objetivo es encontrar una secuencia óptima para la ejecución de unos trabajos y sus respectivas operaciones en las máquinas o recursos disponibles limitados, la optimalidad de la secuencia depende de la función objetivo. Es un proceso de toma de decisiones que constituye uno de los problemas más importantes en gestión de la producción, tanto desde el punto de vista teórico como práctico [2]. Las investigaciones recientes en la teoría de la secuenciación

de operaciones han sido objeto de significativos desarrollos con técnicas como reglas de despacho altamente sofisticadas y algoritmos basados en heurísticas y meta heurísticas, estos desarrollos han sido formulados por un amplio y diverso espectro de investigadores, indicando entonces, la multiplicidad de este campo, además el amplio campo de aplicación de la secuenciación, a su vez se han hecho acercamientos a diferentes tipos de configuraciones de planta.

Uno de los más populares modelos en la teoría de la secuenciación es el problema del Job Shop (JSSP), que es considerado como una buena representación de las configuraciones de planta industriales reales modernas; sin embargo en investigaciones recientes se estableció un problema más genérico para el JSSP, llamado el problema del Flexible Job Shop Scheduling (FJSSP) por lo que es una representación realmente aproximada a los sistemas de manufactura actuales, dado la posibilidad de la configuración de centros de trabajo en la planta, que constituye un conjunto de máquinas que pueden ejecutar una de las operaciones específicas de los trabajos, por lo que el FJSSP es más complejo que el JSSP, por que la meta de la secuenciación es, escoger una asignación para cada una de las operaciones de los trabajos a alguna máquina del conjunto, el problema genérico del JSSP ha sido menos estudiado que este último, sin embargo se ha conocido como un problema NP-Hard, por lo que los investigadores han desarrollado diferentes técnicas para llegar a acercamientos exitosos en la secuenciación del problema Flexible Job Shop Scheduling. Por lo tanto, El objetivo de este documento es ofrecer una visión de este problema de secuenciación y los métodos, desde sus inicios hasta la actualidad, destacando las referencias y las propuestas de cada autor.

II. CONCEPTOS BÁSICOS

Según Fogarty, Blackstone y Hoffmann [1991] plantean que, la planeación de producción es la fijación de objetivos de producción cuantificables en el tiempo (Planeación estratégica) y la toma de decisiones sobre la determinación de cómo lograrlos; por lo que esto implica determinar las cantidades a producir para un horizonte temporal de planificación, conformando lo que se denomina un plan de producción [3].

La programación de operaciones (scheduling), en un sentido amplio, puede pensarse como; la asignación de recursos (máquinas) en un lapso de tiempo para realizar un conjunto de trabajos [4], o bien como resolver el problema de encontrar la asignación óptima de ciertos recursos a determinados tareas [5].

Por lo que, en un problema de planificación siempre existirá tres componentes diferenciadas: las tareas u operaciones (*trabajos*) que se tienen que realizar, los recursos disponibles para su realización (*máquinas*), y las finalidades u objetivos (*función objetivo*) que se desean lograr y que nos permite identificar, entre varias planificaciones posibles, aquellas que

sean óptimas, de acuerdo a los componentes anteriores, los problemas de planificación se dividen en dos, *determinísticos* y *estocásticos*.

Se denomina *determinístico*, (*"Deterministic machine scheduling"*) cuando todos los datos del problema de planificación son conocidos a priori. Estos modelos son estudiados por la optimización combinatoria; una característica común a la mayoría de los problemas estudiados por la optimización combinatoria es que son relativamente "fáciles" de plantear, pero difíciles de modelizar y, consecuentemente, mucho más difíciles de resolver, [6].

Una extensión natural de los modelos de planificación determinístico, consiste en asumir que ciertos datos del problema varían aleatoriamente y, de esa forma, aparecen los problemas de Planificación Estocástica (*"Stochastic machine scheduling"*) [7].

El estado de planificación determinística, es especialmente frecuente en los problemas de planificación y secuenciación de tareas, por tanto su resolución exige el uso de medios de computación adecuados. Los correspondientes aspectos computacionales se estudian analizando la *complejidad computacional* de los métodos propuestos.

En este sentido, es importante mencionar el análisis de la *complejidad computacional* de problemas combinatorios y las implicaciones resultantes para el diseño y análisis de algoritmos adecuados. Se acepta comúnmente que un problema está bien resuelto o es fácil si se puede resolver por un algoritmo cuyo tiempo de ejecución éste acotado por una función polinomial en el tamaño del problema [8]. En la mayoría de problemas de optimización combinatoria no se tiene conocimiento de la existencia de tal algoritmo; entonces, surge la cuestión de probar si el problema es *NP-Hard* o puede resolverse en tiempo polinomial, (los problemas NP-hard son un subconjunto de la clase NP -Problemas para los que no se puede tener una solución en tiempo polinomial para todas sus instancias- Con la característica de que todos los problemas de ésta clase pueden ser reducidos a NP). Los problemas para los que se puede encontrar un algoritmo de solución en tiempo polinomial forman la clase P, que es un subconjunto de la clase NP, [9 y 10]. Es decir, el tiempo de cómputo que se requiere para resolver uno de estos problemas se incrementan conforme crece el tamaño del problema presentando una dependencia funcional tal que no admite ser acotada por un polinomio; en consecuencia a esto, la resolución de este tipo de problemas se centra en encontrar soluciones cuasi -óptimas en tiempos de aplicación razonable; lo que quiere decir que se prefiere comprometer la calidad de la solución a cambio de obtener el acceso del resultado en un tiempo viable.

Muchos autores han medido los algoritmos de acuerdo a varios indicadores, para escoger el mismo, uno de los distintos tipos de indicadores, incluyen, la raíz cuadrada del retraso, el tiempo medio de estancia en el sistema y el número de cambios

de recursos auxiliares. A medida que aumenta la utilización de la planta, el estudio revelará que un algoritmo escogido proporciona muy buenos resultados en los tres tipos de indicadores mencionados [11].

Teniendo en cuenta que los algoritmos adaptativos para la programación de la producción, como los algoritmos genéticos han entregado buenas soluciones durante el proceso de investigación, y usualmente mejoran al acercamiento de la solución óptima, ese es el enfoque que se discutirá a través de este documento.

Los *algoritmos genéticos* (AG) son algoritmos de optimización aproximados que utilizan una terminología basada en los procesos de evolución genéticos de las especies. En realidad, los AG emplean los mecanismos de selección natural y la genética para encontrar soluciones óptimas a problemas de optimización combinatorios. Los AG surgieron a partir de los estudios llevados a cabo por John Holland en la Universidad de Michigan. Goldberg [1989] proporciona una interesante recopilación del trabajo práctico llevado a cabo en esta área.

Los Algoritmos Genéticos pueden entenderse como una generalización de los algoritmos estocásticos de búsqueda local. El espacio de búsqueda de las soluciones candidatas se asemeja a un conjunto de organismos agrupados en poblaciones que evolucionan en el tiempo. La función objetivo del problema de optimización se implementa como una función de adaptación de organismos. La función de adaptación representa el grado de adaptación al entorno del organismo. Los Algoritmos Genéticos se aplican a campos como la planificación de operaciones en planta, problemas de corte de materiales y de componentes irregulares, el entrenamiento de redes neuronales y muchos otros problemas de optimización. Los AG presentan muchas ventajas de cara a su empleo en problemas de programación de operaciones en planta. La calidad de la solución y el tiempo de proceso dependen directamente de los parámetros de los AG, concretamente del número de organismos de la población y del máximo número de generaciones a probar. Esta característica les confiere gran flexibilidad. Rossi y Dini [2000] [12] afirman que es posible encontrar los mejores valores de los parámetros de configuración de un AG a través de otro procedimiento de optimización.

III. DESCRIPCIÓN DEL PROBLEMA

El problema de secuenciación de FJSSP presenta una serie de variantes dependiendo de la naturaleza y el comportamiento; tanto de las operaciones como de las máquinas. Una de las variantes más difíciles de plantear, debido a su alta complejidad computacional, es aquella en donde las tareas son dependientes y las máquinas son diferentes. En esta variante cada trabajo presenta una lista de operaciones que la preceden y para ser

ejecutada deben esperar el procesamiento de dicha lista en su totalidad. A esta situación hay que agregarle la característica de heterogeneidad de las máquinas: cada tarea demora tiempos distintos de ejecución en cada máquina. El objetivo será minimizar el tiempo acumulado de ejecución de las máquinas, conocido en la literatura como makespan

En el problema de asignación de operaciones en la configuración de planta tipo Flexible Job Shop, presenta dos problemas a resolver; el primero consiste en la asignación de cada operación a la máquina y el segundo es referente a la secuencia de operaciones a cada máquina para minimizar la función objetivo.

El problema de FJSSP suele ser definido por las siguientes condiciones [13]:

- Hay n trabajos con subíndice i , y estos trabajos son independientes entre ellos.
- Cada trabajo i tiene una secuencia de operación, denotada por J_i .
- Cada trabajo consiste de una o más operaciones O_{ij} .
- Cada secuencia de operación está ordenada por un juego de operaciones O_{ij} .
- Hay m máquinas con subíndice K (El k -ésima máquina es denotado por m_k)
- Para cada operación O_{ij} , hay un juego de máquinas capaces de cumplir con la función objetivo. Este juego de máquina es denotado por U_{ij} .
- Los tiempos de procesamiento P_{ij} , de una operación O_{ij} en una máquina K es predefinido y mayor que cero.

Restricciones generales

- Cada operación no puede ser interrumpida durante el cumplimiento de la ejecución de la misma.
- Todas las máquinas están disponibles en el tiempo $t=0$
- Cada máquina K , no puede procesar más de una operación simultáneamente.

Un problema de programación de operaciones está definido por tres campos α, β, γ , donde α representa la configuración del taller, β representa las restricciones y/o características propias a las tareas, y γ es la función objetivo que se va a minimizar.

Por consiguiente, nuestro problema es denotado por Vaessens, en su tesis doctoral, [1995], basada en los estudios de Graham y otros [1966], entonces, el problema puede ser denotado como [14]: $m1m/Chain, opd''\mu / Cmax$, donde: el primer campo m especifica el número de máquinas, es una constante, 1 especifica que alguna operación requiere al menos de una máquina para ser procesada y la segunda m indica un límite superior del número de máquinas que puede procesar una operación. El segundo campo muestra el estado de las restricciones de precedencia y el máximo número de operaciones por trabajo, mientras que el tercer campo indica la función objetivo.

Este problema es conocido como fuertemente NP-Hard, incluso si el trabajo tiene tres operaciones y tan sólo dos máquinas [15]. Williamson y otros [16] demostraron que, cuando el número de máquinas, trabajos y operaciones por trabajo son parte de la entrada, en este caso no existe una aproximación de tiempo polinomial con el peor de los casos como límite inferior de $5/4$ a menos que $P=NP$. Otra versión de la complejidad computacional es presentada por González y Sahni [1978] es que el problema es fuertemente sensible cuando $m=3$ y $\mu=3$ [17]

IV. SECUENCIACIÓN DE OPERACIONES DE UNA PLANTA CON CONFIGURACIÓN TIPO JOB SHOP

El problema clásico de programación Job Shop para minimizar el makespan, $J//C_{\max}$ fue presentado por primera vez por Fisher y Thompson, [1963]; desde entonces el problema ha sido un tema estándar en la programación y secuenciación de operaciones. [18]; Morton y Pentico, [1993] Y Rinnooy Kan, [1976]; proporcionaron un estudio para este problema de la demostración de que el problema es NP- Hard.

El problema del Job Shop es un programa que consiste en; un juego de N trabajos en M máquinas, cada trabajo i es procesado en una máquina de acuerdo a la secuencia de operaciones J_i , además la secuencia final de las operaciones es independiente de los tiempos de procesamiento de cada una de las operaciones y los trabajos; donde generalmente se ha encontrado que las investigaciones determinan como criterio de optimalidad para este problema, la minimización de la función regular de optimización, Makespan.

Uno de los acercamientos más tempranos al esquema de algoritmo genético fue desarrollado por Nakano y Yamada [1991], quienes crearon una *codificación binaria basada en las relaciones de precedencia de operaciones con iguales máquinas; una estrategia llamada "Forcing"*, en 1992 estos mismos autores mejoraron este trabajo, definiendo un operador de cruce llamado GA/GT basado en el algoritmo desarrollado por Giffler y Thompson [1960]. En el desarrollo de Nakano y Yamada los cromosomas están en una lista ordenada de tiempos de completación de las operaciones [19].

Un acercamiento importante mediante algoritmos genéticos lo dio Bean [1994], donde propuso sistemas de aleatorios para la codificación de esta meta-heurística, Bean y norma [1997] desarrollaron y mejoraron el desarrollo anterior de Bean, donde establecieron cada gen, mediante una generación aleatoria de caracteres, que se componía de dos partes: una parte entera $\{1, 2, \dots, m\}$ y una segunda fracción generada de un aleatorio $(0,1)$. La parte entera del gen es la asignación de la máquina mientras que la parte binaria, determina la secuencia de operación en cada máquina [20].

Kumar y otros [1996] [21] direccionaron su trabajo de investigación, en el desarrollo de un *algoritmo genético y una*

combinación de reglas de despacho, el propósito de este algoritmo fue mejorar una solución anteriormente determinada para este problema, reduciendo el makespan al 30% sobre ese sistema, que consistía en, un algoritmo ramal y límite que determinaba el óptimo makespan probado para un pequeño grupo de problemas y basado en heurísticas de reglas de despacho específicas para minimizar diferentes medidas de cumplimiento disponibles para un gran grupo de problemas;

Azarmi y otros [2000] [22], ellos propusieron un *algoritmo genético con una representación incompleta*, lo que quiere decir es que, el número de genes es menor que el número de operaciones y lo aplicaron al JSSP, lo más importante de su desarrollo fue la decodificación en la secuenciación del cromosoma y el resto del programa es completado por una simple regla heurística, este enfoque fue considerado debido a que; la solución de este tipo de problemas es contenido por un cromosoma a una programación real, desde el primer gen hasta el último gen; presentándose dos problemas para los algoritmos genéticos, alta redundancia y pequeña significancia de posteriores genes sobre la calidad de la secuenciación.

Eikelder y otros [2001] [23], ellos describen un *algoritmo híbrido, una combinación de algoritmos genéticos y búsqueda tabú*, los componentes de su algoritmo genético son: Una representación natural de la solución y una recombinación capaz de transmitir las características significativas de los padres a los hijos; los problemas de factibilidad y restricciones de los trabajos fueron discutidas y solucionadas en el esquema de trabajo. El operador de cruce (Recombinación) desarrollado preserva la diversidad en la secuencia de los padres. Además, todo el algoritmo genético fue hibridizado con búsqueda tabú (tabu search, TS) y sus resultados experimentales mostraron que es más beneficioso combinar GA con TS que otros acercamientos como el recocido simulado, porque requiere menos tiempo para obtener una solución de calidad y una gran diferencia en el orden de las magnitudes en la solución.

En estas instancias los autores requieren determinar los valores para los parámetros de los algoritmos propuestos para entregar en menor tiempo una solución de calidad, por lo que, la parametrización de los modelos comenzó a ser un tema adicional en los trabajos desarrollados para la secuenciación de operaciones, cómo método para mejorar la calidad de la solución y los tiempos computacionales; Pérez y Herrera [2006], [24] desarrollaron un *algoritmo genético multimodal, para el problema de Job Shop, con el criterio de optimalidad del makespan*, donde inicialmente pretendían conocer más sobre este método, el que a demostrado ser uno de los más potentes, y conocer sobre el comportamiento de este método con respecto a la variación de sus parámetros; donde analizaron la influencia de dichos parámetros en la eficacia del método, y principalmente, sobre su efectividad, entendida esta como la capacidad de alcanzar en una única ejecución del algoritmo el mayor número posible de soluciones óptimas globales del problema. Además, estudiaron la posibilidad de dirigir la

búsqueda hacia la exploración del espacio de búsqueda o su explotación, variando la parametrización, la parametrización del algoritmo lo desarrollaron mediante un método denominado *clearing*.

En el 2007, [25] desarrollaron un *algoritmo basado en agentes, colonia de hormigas para el problema del Job Shop scheduling para minimizar el makespan*, donde los parámetros del algoritmo fueron calibrados utilizando un diseño experimental, factorial completo; en los modelos de secuenciación los parámetros son calibrados para reducir los tiempos computacionales de ejecución de algoritmo sin sacrificar la calidad de la solución. Y para determinar los efectos e interacciones en el valor del makespan, utilizaron un diseño experimental fraccionado.

En el 2008, [26], consideraron el problema de Job shop con fecha prevista para iniciación de cada uno de los trabajos y fecha prevista de entrega, con el objetivo de *minimizar el peso total de la tardanza*, en esta investigación usan un algoritmo genético combinado con una búsqueda interactiva local, que usa una larga ruta de acercamiento en un grafico de un modelo disyuntivo, donde es empleado el diseño de experimentos para calibrar los parámetros y operadores del algoritmo; donde ellos detectaron que, los algoritmos genéticos son altamente dependientes de las cromosomas que son decodificadas, también ellos muestran que, la eficiencia de los algoritmos genéticos no depende de la construcción de la programación cuando una heurística de búsqueda local interactiva es usada.

Sin embargo, en las investigaciones recientes los acercamientos a la configuración de planta tipo Job Shop han sido, mediante optimización bicriterio, es decir dos criterios de optimización, donde la función objetivo no es una función regular. En el 2008, se hicieron acercamientos con optimización bicriterio (Makespan y máxima demora) [27] en esta investigación encontraron una aproximación a la frontera de Pareto. Ellos propusieron un algoritmo genético rápido y elitista, basado en NSGA-II para solucionar el problema, en este trabajo la población inicial del algoritmo es generado por un aleatorio de partida generado usando un algoritmo de búsqueda tabú; los dos algoritmos desarrollados son probados mediante programa benchmark y los resultados computacionales muestran la efectividad y eficiencia del algoritmo por la resolución de ambos métodos.

V. SECUENCIACIÓN DE OPERACIONES PARA LA CONFIGURACIÓN DE PLANTA TIPO FLEXIBLE JOB SHOP

El problema del Flexible Job Shop Scheduling (FJSSP) es una generalización del clásico problema de Job Shop, donde se determina y se optimiza de acuerdo a un criterio, las rutas de los trabajos, por lo que significa decidir, la ruta para cada operación, en que máquinas ha ser procesadas y la disponibilidad de las mismas, optimizando esa secuencia.

Varios procedimientos heurísticos (reglas de despacho, búsqueda local y meta-heurísticos) han sido utilizado en las investigaciones de acercamiento a la configuración de este tipo de planta como son: búsqueda tabú (tabu search, TS), enfriamiento simulado (simulated annealing, SA) y algoritmos genéticos (genetic algorithm, GA) han sido desarrollados en los últimos años para FJSSP. La solución de este tipo de problemas pueden ser clasificados en dos categorías: *planteamientos jerárquicos y planteamientos integrados*.

A. Planteamiento jerárquico e integrado para el enfoque de solución

El *planteamiento jerárquico*, trata de resolver el problema por descomposición de una secuencia en subproblemas, y con ello reducir la dificultad. Una típica descomposición es la asignación-luego-secuenciar, partiendo desde la observación trivial, una vez que la asignación es hecha, el resultado de la programación es un Job Shop Scheduling. Este enfoque ha sido trabajado por Brandimarte P. [1993], Paulli J. [1995], Chambers JB y Barnes JW. [1996]; ellos tratan de resolver el problema de asignación utilizando reglas de despacho, y luego resolver el resultado de JSSP usando diferentes algoritmos, entre ellos búsqueda tabú [28].

Por otra parte el *planteamiento integrado*, es realizar de forma integrada la asignación y la secuenciación, posee más dificultad para ser resuelto, pero en general da mejores resultados, como lo describe Vaessens y otros [1994], Hurink J, Jurish B, Thole M., [1994], Mastrolilli M y Gambardella LM., [1996] y Dauzère-Pérés S y Paulli J., [1997]. Ellos adoptan un planteamiento integrado, proponiendo diferentes formas de búsqueda tabú para resolver el problema [29]. Mastrolilli y Gambardella, [30] muestran resultados computacionales los cuales enseñan que el desempeño de la búsqueda tabú es mejor que otros desarrollos heurísticos, ambos en términos de tiempo de procesamiento computacional y calidad de la solución.

B. Enfoque de solución para el problema General del Job Shop

Muchos autores han definido el modelo general y han trabajado con el en sus investigaciones, el problema que han considerado como modelo *General del Job Shop Scheduling Problem* (GJSSP), es definido como un numero de trabajos a ejecutar, cada uno con un juego de operaciones que deben ser procesadas en un específico orden; las rutas no son necesariamente lineales; los recursos son máquinas que pueden ser usadas para diferentes tipos de procesamiento y la ejecución de cada una de las operaciones puede ser realizada en un recurso elegido de un grupo de ellas. Este problema es considerado por unas razones prácticas, no necesariamente a los autores les interesa un único criterio de optimalidad, sino que requieren varios criterios para tener una mejor calidad en la solución [31], en la mayoría de los trabajos relacionados a la

programación de tareas se concentra en el JSSP, son pocos los que hay sobre GJSSP, que puede ser considerado plenamente como un FJSSP, los siguientes son los trabajos desarrollados en la literatura que solucionan el problema del GJSSP.

Para el GJSSP, Paullí [1995] [32] plantea un algoritmo jerárquico donde el primer paso es resolver el problema de asignación, y el segundo paso es resolver el JSSP. Dauzère-Pérés S y otros [1997] [33] proponen un algoritmo de búsqueda tabú para resolver el makespan de un GJSSP. Kacem y otros [2002] [34] plantea la hibridación de un algoritmo evolutivo y lógica difusa. Alvarez-Valdez y otros [2005] [35] plantean un algoritmo heurístico para un GJSSP para una industria de vidrio con restricciones de no espera o traslapes. Con el objetivo de encontrar una secuencia con un criterio basado en las penalizaciones de tardanza y en la prontitud. Los autores generan un algoritmo de dos pasos; en el primero priorizan las rutas de los trabajos que son utilizadas para resolver el problema, y en el segundo paso mejoran la solución con una búsqueda local.

Para la investigación de G. Vilcot y J. Billaut, [2008], planteada anteriormente en el cual desean comparar los procedimientos de búsqueda tabú y el algoritmo genético, consideran como problema base el *General Job Shop Scheduling Problem* GJSSP.

C. Enfoque de solución Jerárquico para el Flexible Job Shop mediante el enfoque de máquinas paralelas

Al procesar una cantidad de trabajos estos deben pasar por varias etapas, y en cada etapa hay un determinado número de máquinas paralelas con diferentes capacidades de procesamiento de los trabajos. Además generalmente hay etapas que representan cuellos de botella para el procesamiento de los trabajos, con lo cual al secuenciar los trabajos se tienen en cuenta las restricciones, como la capacidad y la productividad de cada máquina, el tomar la decisión de la posición de los trabajos para las máquinas y la selección de los tiempos de comienzo de cada trabajo, es un proceso complejo que requiere ser analizado y discutido desde varias perspectivas, el problema de secuenciar en una planta con configuración FJSSP, puede ser adaptado a secuenciar, teniendo en cuenta principalmente la configuración de los recursos en cada etapa.

Chichang [2005] [36], realizó una investigación evaluando un algoritmo genético con partición de genes sub-indexados con número distintivo (GASP), con operadores que incrementan la diversidad de permutación, cuando el número de generaciones con la misma mejor solución excede un punto de inicio, un comportamiento es cambiado para ayudar a que el algoritmo se impulse fuera del sub-óptimo y con ello prevenir una convergencia prematura; utilizando el método de la selección de la ruleta, encogiendo la base de cromosomas sobre el valor de la función, y asignando a los cromosomas una

distribución de probabilidad a favor de los buenos cromosomas, y con ello obtener una mejor oportunidad de generar una siguiente generación buena. Considerando los factores en la evaluación de la programación; la tardanza, la prontitud, los tiempos de utilización y la variable de costos es optimizada la secuencia para las máquinas en paralelo.

Además, en la programación de los trabajos en las plantas de manufactura, se utilizan métodos de secuenciación heurísticos, los que se basan en reglas y métodos adaptativos. Los heurísticos basados en reglas se han realizado dependiendo de los tipos de ordenamiento y los tiempos de carga de las máquinas, con lo cual las reglas que más frecuentemente se utilizan para configuraciones con una máquina, como lo muestra la literatura [37] al secuenciar a partir de Earliest Due Date (EDD), y [38] con un Shortest Processing Time (SPT). Para extender estas reglas para máquinas en paralelo necesitaron integrar las características de los trabajos asignados a las máquinas. Por otra parte se han hecho desarrollos mediante los heurísticos adaptativos en estos desarrollos se encuentran autores como: Aarts y Korst, [1989], Van Laarhoven y Aarts, [1987], Palmer, [1996], utilizando algoritmos de enfriamiento simulado, otros utilizan búsqueda Tabú entre ellos, Brandimarte y Calderini [1995], Ben-Daya y Al-Fawzan, [1998], Armentano y Ronconi, [1999], además de algoritmos Hill Climbing se encuentran Bonet y Geffner, [1999] y en secuenciación con algoritmos genéticos se encuentran autores como: Goldberg, [1989], Gen y Cheng, [1997], [Liu y Tang, 1999], [Lee et al., 2002], [Kacem, 2003]. Los cuales se han acercado a la solución óptima a través de la interacción de parámetros ajustados para obtener una solución subóptima. Una de las características de los algoritmos adaptativos es que todos ellos necesitan alguna evaluación o juzgamiento de que tan buena es la solución subóptima. Ellos pueden luego modificar la configuración de probabilidad existente generando soluciones muy similares, y continuar la búsqueda de la solución óptima hasta que resulte satisfactoria [39].

D. Enfoque de solución utilizando Algoritmos Genéticos

Recientemente los *algoritmos genéticos* han sido adoptados para resolver el problema de Flexible Job Shop Scheduling (FJSSP), generando un incremento en el número de estudios desarrollados como lo muestran las publicaciones, por que los algoritmos genéticos tienen unos atributos que permiten desarrollar adecuadamente la tarea de la secuenciación, entonces, los GA tienen cinco atributos básicos: representación genética de soluciones para los problemas; el camino para crear la población inicial; operadores genéticos para cambiar la población genética; función de evaluación y valor de los parámetros; estos parámetros han permitido que los investigadores desarrollen teorías a través de estos elementos y mejoren los acercamientos desde el modelo simple de algoritmo genético para optimizar la secuencia de operaciones mediante alguna medida de cumplimiento en la producción.

Algunos de los trabajos más relevantes los han desarrollado Lehmann y otros [1999] [40], ellos proponen un algoritmo genético para el FJSSP, donde dividen la representación de cromosomas en dos partes, la primera define la política de enrute, y la segunda los operadores de secuencia sobre la máquina; según ellos, la principal ventaja de su algoritmo es que puede resolver una larga clase de problemas, incluyendo el JSSP; las pruebas iniciales del algoritmo muestran que encontraron una alta calidad en la secuenciación para los problemas de una talla realista y entonces consideran que es una prometedora aplicación práctica. El algoritmo presentado sólo considera el criterio del Makespan y dentro de ese mismo algoritmo pueden considerar otro criterio como es el peso del flujo de tiempo total o flujo de tiempo total o la tardanza total; lo que ellos consideraron como debería ser un trabajo futuro en el desarrollo obtenido.

Kacem I, Hammadi S, Borne P., [2002], Jia HZ, Nee AYC, Fuh JYH, Zhang YF., [2003] y Tay JC y Ho NB [2004], todos ellos han utilizado *enfoques integrados*, cada uno con diferentes esquemas de representación, generación de población inicial, selección de cromosomas y estrategias de generación de poblaciones.

[41] Jeong y otros [2002], evalúan las alternativas de procesos en la planta para trabajos con restricciones de precedencia en la planeación avanzada y secuenciando sistemas utilizando algoritmos genéticos, incluyendo en la función de evaluación parámetros de intervalos de entrega de los trabajos y costos de salvamento de outsourcing. Kacem I, y otros [2002] [42], utiliza la representación de cromosomas que combina la información de secuencia y de enrute, y desarrolla un planteamiento por localización para encontrar una asignación inicial prometedora. Las reglas de despacho son luego utilizadas para la secuencia de las operaciones. Una vez que la población inicial es encontrada, ellos aplican operadores de mutación y de cruce juntando y modificando la asignación y la secuencia, produciendo mejores individuos para cada generación

Kacem, [2003] [43] utiliza un algoritmo genético con operadores de preservación de precedencia para minimizar el makespan y los tiempos de carga de las máquinas. Ese mismo año este mismo autor presenta, un algoritmo genético para el peor caso de análisis [44], donde propone un método heurístico para solucionar el problema, mediante operadores de preservación de precedencia, para una optimización multiobjetivo, además usando un método de localización en el primer paso, en el segundo paso un algoritmo evolucionario para encontrar la solución inicial, tomando en cuenta todas las restricciones y especificaciones del problema, en este proceso muchos operadores son asignados, concernientes a dos componentes del problema: La asignación y la secuencia. Algunas pruebas de simulación realizadas en esta investigación muestran que, la pequeña distancia entre los límites inferiores y el valor del criterio obtenido por la solución generada, es

generalmente satisfactoria y promisoria, en su trabajo concluyen que sus resultados obtenidos demuestran la buena calidad de los diferentes límites y la adecuación del método evolucionario, al problema.

Sankar y otros, [2003] [45] adopta un algoritmo genético para minimizar el costo total de penalización y maximizar la utilización de las máquinas para un sistema de manufactura flexible.

Tay J.C y Ho N.B., [2004] [46], propone una metodología eficiente llamada GENACE basada en la cultura de arquitectura evolucionaria para resolver el FJSSP con recirculación, muestran como un CDR (Composición de reglas de despacho) son usadas para solucionar el FJSSP con recirculación por ellos mismos y generando un mecanismo de inicialización GENACE. Ellos adoptaron una arquitectura cultural evolucionaria, manteniendo el conocimiento del esquema y la localización de los recursos aprendidos sobre cada generación. La creencia de la influencia de los espacios de mutación y selección sobre una representación factible del cromosoma. Los resultados experimentales que muestran de GENACE obtienen mejores resultados para los límites superiores para 11 de 13 problemas benchmark con una mejora de los factores entre un 2 y 48 por ciento cuando los resultados son comparados con los estudios realizados por Kacem y Brandimarte y el uso de CDR solamente.

Tanev, y otros [2004] [47], proponen un algoritmo híbrido evolucionario, el cual combina las prioridades de reglas de despacho (Priority Dispatching Rules) PDRs con algoritmos genéticos; con las reglas de despacho determinan cual es el orden a ejecutar de la próxima tarea, desde la lista de trabajos no secuenciados. Comparando las otras alternativas, las PDRs ofrecen la ventaja de la simplicidad, generando un bajo costo computacional y pudiendo aplicar a los problemas reales que se presentan en las plantas de máquinas de inyección de plástico (factories of plastic injection machines FPIM) que dan lugar a la configuración de planta tipo Flexible Job Shop FJSSP, considerando que las empresas tienen recursos (máquinas, moldes, etc.) que son impredecibles, el algoritmo influye de forma importante en el éxito de la secuenciación.

El algoritmo planteado no trata de predecir el futuro sino que toma decisiones según el presente, con lo cual localiza los procesos durante la secuenciación y consecuentemente reduce el costo computacional de un algoritmo, al descomponer las tareas de la programación de los trabajos en agentes autónomos donde cada agente es una máquina de la planta. Sin embargo, el procedimiento de PDR tiene elementos no ventajosos como: Su corta visión, utilizando un óptimo local que está lejos de un óptimo global, ningún PDR simple puede aplicarse para posibles casos de secuenciación de FPIM. Esto se debe a que se necesitaría un desarrollo empírico que involucre PDRs como combinación de ellos, que se podría en instancia concreta secuenciación de FPIM

Por lo anterior, se observa que los PDRs tienen desventajas que deterioran una solución que se acerque a un óptimo global, de forma que la generación de una combinación de un PDRs y de un algoritmo genético (GA) utiliza las ventajas de ambos procedimientos dando lugar a un procedimiento de mejor calidad.

Borne y otros (2004) [48], desarrollaron una aplicación de *algoritmos evolucionarios para el FJSSP con restricciones de problemas del mundo real*, ellos demostraron que escogiendo una conveniente representación del cromosoma (Codificación en paralelo) que es un paso importante según ellos, para encontrar mejores resultados, desarrollaron un operador genético adaptado para cada representación (Cambiar y asignar la mutación para el PME (Parallel machina encoding) y controlar la mutación para el PJE (Parallel Jobs encoding)) y un eficiente creación de una población inicial, mediante una combinación de algunos métodos para PME y un aleatorio acercamiento para el PJE porque todas las restricciones son integradas en la sintaxis del cromosoma. Una conveniente selección de los parámetros del genético para una aplicación de los algoritmos evolucionarios. Los parámetros (Tasa de cruce, tasa de mutación, tamaño de población, etc) son usualmente seleccionados heurísticamente. En este trabajo no hay líneas exactas de estrategia a ser adoptadas para los diferentes problemas. En este trabajo aplican claramente, un tamaño de población fija con diferentes valores de tasa para el cruce y mutación, ellos encontraron que la mutación controlada reduce la selección hecha a ciegas. Sus resultados de simulación mostraron las representaciones paralelas propuestas son más convenientes para el FJSSP, confirmando la efectividad de los acercamientos propuestos.

Gen y otros [2005] [49], ellos proponen una nueva *base de multiestado de representación para simplificar el cromosoma*, usando este método los acercamientos desarrollados para los cruces y la mutación pueden ser aplicados, la importancia de este desarrollo se da, porque la efectividad de la solución es principalmente afectada por la complejidad de la representación del cromosoma; el algoritmo desarrollado fue diseñado para tres criterios de optimización incluyendo el Makespan, carga de trabajo total para cada una de las máquinas y la máxima carga de trabajo para todas las máquinas, mediante pruebas numéricas experimentales ellos demostraron la eficiencia de este acercamiento a la solución del FJSSP.

Chan y otros [50], también exploran un *algoritmo genético, basado en el acercamiento a los recursos restrictivos de asignación de operaciones a máquina*, que a su vez puede resolver interactivamente el problema genérico del JSSP, es decir, el problema resuelto mediante este esquema de acercamiento es RCFJSSP. En este estudio el desarrollo es llamado IGA y muestra las mejoras a este ambiente y al ambiente de las múltiples máquinas bajo recursos restrictivos, así, demostraron que los conflictos entre el Makespan y MIC (Machina idle cost) pueden ser solucionados bajo diferentes niveles de recurso RL; de igual forma, este acercamiento puede

ser fácilmente modificado a tipo II FJSSP con múltiples máquinas no idénticas bajo recursos restrictivos.

Gen y otros, [2007] [51], realizan la programación del problema para planta tipo FJSSP teniendo en cuenta tres objetivos: minimizar el makespan, minimizar el máximo tiempo de carga de maquina y minimizar el tiempo total de tiempo de carga, desarrollando un *algoritmo híbrido genético (hGA)* para el problema. Basados en la habilidad de cruce y mutación de los vectores, mejorando la población y la búsqueda por medio de una descendencia variable de vecindad (VND), desarrollando un método eficiente para encontrar intervalos de tiempo asignables para la eliminación de operaciones, basados en el concepto de prontitud y tardanza de un tiempo presentado. El movimiento óptimo local de una operación es mejorado mediante el movimiento de dos operaciones simultáneas. Como se ha descrito, en los trabajos desarrollados se ha encontrado que los autores en las diferentes configuraciones de planta especialmente este tipo de configuración, consideran restricciones, ya que una programación ideal ocasionaría que la producción y el Scheduling no se ejecuten como se plantea [52]; existen varios *tipos de restricciones* en la planta, los cuales se definen en dos *tipos fijos y No fijos*. Las restricciones asociadas a los tipos de *restricciones fijos*, son aquellos que empiezan y terminan en un tiempo definido; mientras que en las *restricciones No fijas*, no existen tiempos definidos de comienzo y fin de la parada, periodo en el cual las máquinas no se encuentran disponibles, estos tiempos deben ser determinados durante la secuencia de producción.

En la producción se presentan mantenimientos preventivos ejecutados una vez, en un intervalo de tiempo. En algunos horizontes de programación no hay actividades de mantenimiento. Además en algunos periodos se presentan una o varias actividades de mantenimiento que deben ser programadas en las máquinas, afectando de forma directa la programación de las mismas. El manteniendo preventivo impone restricciones no fijas de disponibilidad sobre las máquinas y generalmente el tiempo de inicio del mantenimiento preventivo es flexible.

El problema de restricciones no fijas para un Flexible Job Shop Scheduling (FJSSP-nfa), es decir, cada máquina está sujeta a un número arbitrario de tareas asociadas a mantenimiento preventivo PM, para lo cual [53] realizan un algoritmo genético híbrido (hGA), dado que en el algoritmo genético, un cromosoma únicamente contiene información parcial de la solución candidata y la información ignorada es eliminada para la optimización de un método heurístico en el proceso de decodificación. Contrario a los algoritmos genéticos, los operadores de cruce (crossover) y de mutación (mutation) del hGA propuesto son implementados en el espacio del fenotipo, preferiblemente, que al espacio del genotipo de acuerdo al incremento de heredabilidad de un hGA. Dos tipos de vecindades las cuales solamente contienen soluciones que tienen posibilidades de mejorar la solución inicial definida para

el FJSP-nfa, basado en el concepto de ruta crítica. Un procedimiento de búsqueda local que trabaja sobre las dos vecindades es luego hibridado con el algoritmo genético de acuerdo al incremento de la habilidad de búsqueda. Por lo cual generan dos vectores $v1(r)$ y $v2(r)$, donde $v1(r)$ es el vector de asignación de máquinas, y $v2(r)$ es el vector operador de secuencia, y al programar se realizan cambios de las operaciones hacia la izquierda para compactar lo más posible la secuenciación. Un cambio es llamado cambio global izquierdo (global left-shift) si alguna operación puede empezar más temprano en el tiempo sin demorar las otras operaciones aun si el cambio hecho cambia la secuencia de las operaciones. Un programa es semiactivo si no hay cambios locales a la izquierda y por tanto un programa es activo si no hay cambio global a la izquierda. En este trabajo, utilizan la prioridad basada en la decodificación, donde cada operación busca el intervalo de tiempo disponible para implementarlo en su máquina asignada en orden a la representación dada por el vector operador de secuencia.

Estos mismos autores en el 2008, [54] desarrollaron una extensión del trabajo anterior con tres criterios de optimización: Makespan, Minimización de la máxima carga de máquina y la minimización total de la carga. Este acercamiento fue probado computacionalmente con 181 problemas benchmark mostrando el cumplimiento de ese acercamiento, porque muestran que la misma solución es obtenida por la combinación de esfuerzos de trabajos previos, además, encuentran 38 nuevas mejores soluciones.

Otros autores han estudiado como G. Vilcot y J. Billaut, [2008], el desempeño de las diferentes metodologías para problemas particulares, para lo cual [55], comparan un algoritmo Genético con otro de búsqueda tabú, llevando a colación los problemas que se presentan en las empresas de impresión y en la industria de encartonado. El objetivo es *minimizar el makespan y la máxima tardanza*, encontrando una aproximación de la frontera de Pareto. Proponen un algoritmo genético rápido y elitista basados en NGSa-II para resolver el problema. La población inicial es generada aleatoriamente o parcialmente y ambos algoritmos genético y de búsqueda tabú son probados sobre problemas benchmark, para un análisis y comparación computacional para configuraciones de planta Flexible Job Shop obtenidos con ambos métodos.

Pezzella y otros, [2008] [56], toman inicialmente como base de su estudio los desarrollos realizados por Kacem [2002] entonces, ellos desarrollan un algoritmo genético para Flexible Job Shop, en el cual analizan algoritmos que no han sido efectivos para resolver el FJSSP para pequeñas instancias. De hecho los métodos exactos basados en representaciones disyuntivas de los problemas han sido desarrolladas, pero estos métodos no son aplicables para instancias con mas de 20 trabajos y 10 maquinas, Pinedo M., [2002], por lo cual realizan su desarrollo

en algoritmos heurísticos. Desde los estudios anteriores se sabe que ningún algoritmo conocido para FJSSP produce soluciones con una garantía en distancia desde la solución óptima; para mejorar su método, además conducen los resultados experimentales en ejemplos completando el trabajo con un estudio computacional extensivo, para conseguir los anteriores resultados los investigadores recurrieron a una integración de diferentes estrategias para generar la población inicial y las estrategias para reproducción de nuevos individuos

Ho y otros [2008] [57], desarrollaron un estudio para el FJSSP multi-objetivo usando reglas de despacho descubiertas a través de programas genéticos, muestran la evaluación y empleo de parámetros y operadores apropiados para construir reglas de despacho usando genéticos, con un fin hacia la escalabilidad y flexibilidad, donde casi todas esas reglas de despacho son basadas en la experiencia del programador, la generación de estas reglas de despacho se realizaron a través de framework que soluciona efectivamente el JSSP. Cinco reglas de despacho fueron generadas por el programa genético, estas reglas fueron basadas en combinación de parámetros como: tiempos de procesamiento, fecha prevista, fecha de liberación, fecha común, número de operaciones y promedio total de tiempo de procesamiento de cada trabajo. Muchas simulaciones se realizaron para obtener la evaluación de los criterios de optimización de las cinco reglas, variando los grados de flexibilidad del problema; otras cinco populares reglas de despacho de la literatura fueron evaluadas y usadas en la simulación de los problemas benchmark. Los resultados experimentales de esta investigación mostraron que las populares reglas de despacho validadas en la literatura no cumplen satisfactoriamente con todos los criterios, mientras que la combinación de parámetros contribuye eficazmente con las reglas. Encontraron que la EDD es significativamente mejor que otras reglas de la literatura minimizando la tardanza, el flujo de tiempo y el porcentaje de trabajos tardíos mientras que es pobre la minimización del Makespan. Sin embargo, las reglas generadas por el programa genético, cumplieron con los criterios de optimización por encima de la EDD, mostrando esto que, la aplicación desarrollada es efectiva en la construcción de reglas de despacho para solucionar el multi-objetivo FJSSP. Ellos muestran estadísticamente que sus reglas de despacho son suficientes y bien diseñadas, realizando un análisis de varianza (ANOVA). Finalmente, las reglas de despacho desarrolladas pueden ser aplicadas directamente en la práctica sin modificaciones.

VI. CONCLUSIONES Y PERSPECTIVAS

Los procesos de manufactura en la actualidad se han vuelto más complejos por lo que, la secuencia de operaciones se ha convertido en un tema importante para mejorar la competitividad de las compañías, el problema del Flexible Job Shop Scheduling se ajusta de manera adecuada a los problemas reales de casi

todos los tipos de manufactura y en todos los sectores. Entonces, dada la importancia de los problemas prácticos del tipo planeación-scheduling y a su gran dificultad de obtener soluciones por ser del tipo NP-Hard, es de relevancia el dirigir nuestra mirada a los acercamientos a través de meta heurísticas, pues con una investigación más profunda se puede encontrar un camino para una importante ‘simplificación’ en el trato de estos problemas.

De acuerdo a la búsqueda realizada en la literatura encontramos que, los algoritmos genéticos son los más recurrentes a la hora de solucionar el problema genérico del JSSP. Mediante el enfoque de algoritmos genéticos se logra encontrar una alternativa para la evaluación de un problema Flexible Job Shop Scheduling, de acuerdo a la literatura, también encontramos que por medio de este enfoque, se puede incrementar la eficiencia algorítmica en cuanto a complejidad y encontrar buenas soluciones, en cuanto a la asignación y la secuencia.

Dentro de los algoritmos genéticos hay elementos que han sido poco estudiados en la literatura del FJSSP, pero que han sido desarrollados por autores en el problema del JSSP; que son los elementos de la parametrización del algoritmo de solución para el problema, las razones para implementar la parametrización en un modelo es poder obtener numéricamente y de manera significativa, algunos de los efectos, dados por elementos como: el tipo de cruce en la calidad de solución para un problema dado y otro podría ser, la influencia que tiene el tamaño de la población en la calidad de dicha solución. La pregunta ¿cuál es el tamaño idóneo para una población? parece no tener una respuesta clara, dado que intuitivamente se puede pensar que las poblaciones pequeñas corren el riesgo de no cubrir adecuadamente el espacio de búsqueda, mientras que trabajar con poblaciones de gran tamaño puede acarrear problemas asociados al gran costo computacional, por tanto un aspecto importante para desarrollar en trabajos futuros, es la parametrización del enfoque de solución para el FJSSP mediante algoritmos genéticos.

AGRADECIMIENTOS

Los autores quieren presentar los más sinceros agradecimientos a la Dirección de Investigación de la Universidad Nacional de Colombia, Sede Medellín (DIME) por la financiación del proyecto de investigación titulado: “Evaluación de la metodología de algoritmos genéticos con diseño experimental para la programación de plantas con configuración tipo Job Shop” con código No.20201007088 del cual este es uno de sus resultados.

REFERENCIAS

- [1] Dante Boiteux, O., 2007. Estado del arte sobre planificación agregada de la producción. Tesis doctoral, Universidad Politécnica de Cataluña, Instituto de la organización y sistemas industriales: Barcelona.
- [2] Pinedo, M., 1995. Scheduling: theory, algorithms and systems. Englewood cliffs, NJ: Prentice-Hall.
- [3] Londoño, M., 2007. Reprogramación de ordenes de trabajo en el modelo básico, con una fecha de entrega común para minimizar la penalización por prontitud / tardanza, basado en métodos heurísticos. Tesis de pregrado, Universidad Nacional Sede Medellín.
- [4] Op Cit. Pinedo M., [2]
- [5] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. y Shmoys, D.B., 1993. Secuencing and scheduling: Algorithms and complexity. Handbooks in Operations Research and Management Science, North Holland.
- [6] Alcaide López de Pablo, D., 1995. Problemas de planificación y secuenciación determinística: modelización y técnicas de resolución. Curso de ciencia y tecnología. Universidad de la Laguna.
- [7] Op Cit. Alcaide López de Pablo, D. [6]
- [8] Lawyer, E.L., 1976. Combinatorial Optimization: networks and matroids, Holt, Rinehart and Winston, New York.
- [9] Karp, R.M., 1972. Reducibility among combinatorial problems. En: Complexity of computer computations, Plenum Press, New York, pp. 85 - 103.
- [10] Garey, M.R. y Johnson, D.S., 1979. Computers and intractability: a guide of the theory of NP-Completeness, Freeman.
- [11] Álvarez de los Mozos, E. y Zubillaga, F. J., 2004. Análisis comparativo de métodos de secuenciación para la programación dinámica de operaciones. En: VIII Congreso de ingeniería de organización, Lérganes.
- [12] Rossi, A. y Dini, G., 2000. Dynamic Scheduling of FMS using a Real-Time Genetic Algorithm, International Journal of Production Research. Vol.38 (1), pp.1-20.
- [13] Chen H., Ihlou J. y Lehmann C., 1999. A genetic algorithm for Flexible Job Shop Scheduling. En: Proceedings of the IEEE Transactions on Neural Networks, International Conference on Robotics & automation. Detroit, Michigan.
- [14] Jansen K., Mastrolilli M. y Solis-Oba, R., 2005 Approximation algorithms for Flexible Job Shop Problems. En: International Journal of Foundations of Computer Science. Vol 16, pp. 361-379
- [15] Op Cit. Jansen K., Mastrolilli, M. y Solis-Oba, R [14]
- [16] Hall, L., Hoogeveen, J., Hurkens, C., Lenstra, J., Sevastianov, S., Shmoys, D. y Williamson, D., 1997. Short shop schedules. En: Operation Research, Vol. 45, pp. 288 -294
- [17] Gonzalez, T. y Sahni, S., 1978. Flowshop and Job Shop schedules: complexity and approximation. En: Operation research. Vol. 26, pp. 36-52
- [18] Baker, K. R., 1974. Introduction to secuencing and scheduling. John Wiley and Sons, New York.
- [19] Nakano, R. y Yamada, T., 1992. Conventional Genetic Algorithm for Job-Shop Problems, in Kenneth, M. K. y Booker, L. B. En: Proceedings of the 4th International Conference on Genetic Algorithms and their Applications, San Diego, USA, pp. 474-479.
- [20] Norman, B. A. y Bean, J., 1997. Random Keys Genetic Algorithm for Job Shop Scheduling. En: Engineering Design and Automation. Vol. 3, pp. 145-156.
- [21] Kumar, N.S. y Srinivasan, G., 1996. A genetic algorithm for Job Shop Scheduling – A case study-. En: Computer in industry. Vol. 31, pp. 155-160.
- [22] Azarmi, N., Hughest, JG., Song, Y. y Voudourist, C., 2000. A genetic algorithm with an incomplete representation for de Job Shop Scheduling Problems. En: Informe de investigación, University of Ulster at Jordanstown. Intelligent System Research Group.
- [23] Eikelder, Ten., H.M.M., Moraglio, A. y Tadei, I., 2001. Genetic Local Search for Job Shop Scheduling Problem. Technical Report CSM-435 ISSN 1744-8050
- [24] Herrera, F. y Pérez, M. E., 2006. Algoritmos genéticos multimodales: Un estudio sobre la parametrización del método clearing aplicado al problema Job shop. Reporte de investigación, Universidad de Valladolid, TIN2005-08386-C05-01.
- [25] Engin, O., Figlali, Al., Figlali, N. y Özcale, C., 2007. Investigation of ant system parameter interactions by using design of experiments for Job Shop Scheduling problems; Computers & Industrial Engineering, pp. 1-22.
- [26] Dazère-Pérés, S., Essafi, I. y Mati, Y, 2008. A genetic local search algorithm for minimizing total weight tardiness in the Job Shop scheduling problem.

- Computer & Operation research, Vol.35, pp. 2892-2907.
- [27] Billaut J. – C. y Vilcot G., 2008. A tabu search and genetic algorithm for solving a bicriteria Job Shop Scheduling problem, En: Discrete optimization, European Journal of Operational Research, Vol. 190, pp. 398 – 411.
- [28] Ciaschetti, G., Morganti, G., y Pezzella, F., 2008. A genetic algorithm for the Flexible Job shop Scheduling Problem. En: Computers & Operations Research, Vol. 35, pp. 3202-3212.
- [29] Op Cit. Ciaschetti, G., Morganti, G., y Pezzella, F. [28]
- [30] Gambardella, L.M. y Mastrolilli, M., 1996. Effective neighborhood functions for the flexible job shop problem. Journal of Scheduling Vol.3 pp. 3-20.
- [31] Op Cit. Billaut Jean – Charles y Vilcot Geoffrey, [27].
- [32] Paulli J., 1995. A hierarchical approach for the FMS scheduling problem. En: European Journal of Operational Research Vol.86, pp. 32-42.
- [33] Dauzère-Pérés, S. y Paulli, J., 1997. An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search. En: Annals of Operations Research Vol.70, pp. 281-306.
- [34] Kacem, I., Hammadi, S. y Borne, P., 2002. Pareto-optimality approach for flexible job shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic. En: Mathematics and Computers Simulation Vol. 60, pp. 245-276.
- [35] Álvarez-Valdez, R., Fuertes, A., Giménez, R., Ramos, G. y Tmarit, J.M., 2005. A heuristic to Schedule flexible job shop in glass factory. En: European Journal of Operational Research Vol.165, pp. 525-534.
- [36] Chichang, Jou., 2005. A genetic algorithm with sub-indexed partitioning genes and its application to production scheduling of parallel machines. En: Computers & Industrial Engineering, Vol.48, pp. 39-54.
- [37] Cheng, T. C. E., y Sin, C. C. S. (1990). A state of the art review of parallel machine scheduling research. En: European Journal of Operational Research, Vol. 47(3), p. 271-292.
- [38] Cheng, T. C. E. y Chen, Z. L., 1994. Parallel machine scheduling problem with earliness and tardiness penalties. En: Journal of the Operational Research Society, Vol. 45, pp. 685-695.
- [39] Op. Cit. Chen H., Ihlow J. y Lehmann C [13]
- [41] Jeong, C. S., Lee, Y. H. y Moon, C., 2002. Advanced planning and scheduling with outsourcing in manufacturing supply chain". En: Computers and Industrial Engineering, Vol.43(2), pp. 351-374.
- [42] Borne, P., Hammadi, S. y Kacem, I., 2002 Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. En: IEEE Transactions on Systems, Man, and Cybernetics, Part C. Vol.32 (1), pp. 1-13.
- [43] Kacem, I., 2003. Genetic algorithm for the flexible job shop scheduling problem. Proceedings of IEEE international conference on systems, man and cybernetics. Vol. 4, pp. 3464-3469.
- [44] Kacem, I., 2003. Scheduling Flexible Job Shop: A worst case análisis and an evolutionary algorithm. En: International Journal of Computational Intelligence and Applications. Vol. 3, No. 4, pp. 437-452.
- [45] Ponnambalam, S. G., Rajendran, C y Sankar, A. S., 2003. A multiobjective genetic algorithm for scheduling a flexible manufacturing system. En: International of Advanced Manufacturing Technology, Vol. 22(3), pp. 229-236.
- [46] Ho, N.B. y Tay, J.C., 2004. GENACE: An effective cultural Algorithm for solving the flexible job shop schedules, En: Lecture notes in IEEE, pp. 1758-1766.
- [47] Tanev, I. T., Takashi, U. y Yoshihauru M., 2004. Hybrid evolutionary algorithm based real world flexible job shop scheduling problem: application service provider approach. En: Applied Soft Computing, Vol.5, pp. 87-100.
- [48] Borne, P., Hammadi, S. y Mesghouni, K., 2004. Evolutionary algorithms for Job Shop Scheduling. International Journal Application Mathematics computers Science. Vol.14 (1), pp. 91-103.
- [49] Gen, M. y Zhang, H., 2005. Multistage –based genetic algorithm for flexible Job Shop Scheduling Problem. En: Complexity International. Vol. 11, pp. 223-232.
- [50] Chan, F.T.S., Chan, L.Y. y Wong, T.C., 2006. Flexible Job Shop Scheduling Problem under resource constraints. En: International Journal of Production Research. Preview article, pp. 1-20.
- [51] Gao, J., Gen, M. y Sun, L., 2007. Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm. En: Journal Intelligent Manufacture Vol.17, pp. 493-507.
- [52] Schmidt G., 2000. Scheduling with limited machine availability. En: European Journal Of Operational Research, Vol.121, pp. 1-15.
- [53] Gao, J., Gen, M., Sun, L. y Zhao, X., 2007. A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible Job Shop Scheduling Problems. En: Computers & Operations Research Vol. 53, pp. 149-162.
- [54] Gao, J., Gen, M. y Sun, L., 2008. A hybrid genetic and variable neighborhood descent algorithm for Flexible Job Shop Scheduling Problems. En: Computers & Operations Research Vol. 35, pp. 2892 – 2907.
- [55] Op. cit. Billaut J. – C. y Vilcot G. [27]
- [56] Op Cit. Ciaschetti, G., Morganti, G., y Pezzella, F. [28]
- [57] Ho, N. B. y Tay, J. C., 2008. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. En: Computers & Industrial Engineering. Vol 54, pp. 453–473

Universidad Nacional de Colombia Sede Medellín

Facultad de Minas

120 años 
TRABAJO Y RECTITUD

Escuela de Ingeniería de Sistemas

Pregrado

- ❖ Ingeniería de Sistemas e Informática.



Posgrado

- ❖ Doctorado en Ingeniería-Sistemas.
- ❖ Maestría en Ingeniería de Sistemas.
- ❖ Especialización en Sistemas con énfasis en:
 - Ingeniería de Software.
 - Investigación de Operaciones.
 - Inteligencia Artificial.
- ❖ Especialización en Mercados de Energía.

Áreas de Investigación

- ❖ Ingeniería de Software.
- ❖ Investigación de Operaciones.
- ❖ Inteligencia Artificial.

Escuela de Ingeniería de Sistemas
Dirección Postal:
Carrera 80 No. 65 - 223 Bloque M8A
Facultad de Minas. Medellín - Colombia
Tel: (574) 4255350 Fax: (574) 4255365
Email: esistema@unalmed.edu.co
<http://pisis.unalmed.edu.co/>

